JHU/APL
TG 1370
JUNE 1988
Copy No.

Technical Memorandum

A GLR ALGORITHM FOR POLYLINE APPROXIMATION OF WAVEFORMS

R. A. STEINBERG

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION NUMBER(S) JHU/APL TG 1370	5. MONITORING ORGANIZATION REPORT NUMBER(S) JHU/APL TG 1370
6a. NAME OF PERFORMING ORGANIZATION The Johns Hopkins University Applied Physics Laboratory 6b. OFFICE SYMBOL (*If Applicable*) FIF	7a. NAME OF MONITORING ORGANIZATION NAVPRO
6c. ADDRESS (*City, State, and ZIP Code*) Johns Hopkins Road Laurel, Maryland 20707	7b. ADDRESS (*City, State, and ZIP Code*) Johns Hopkins Road Laurel, Maryland 20707-6090
8a. NAME OF FUNDING/SPONSORING ORGANIZATION JHU/APL Independent Research and Development Fund 8b. OFFICE SYMBOL (*If Applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00039-87-C-5301
8c. ADDRESS (*City, State, and ZIP Code*) Johns Hopkins Road Laurel, Maryland 20707	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)

A GLR Algorithm for Polyline Approximation of Waveforms

12. PERSONAL AUTHOR(S)

Steinberg, Richard A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
Technical Memorandum	FROM ____ TO ____	June 1988	76

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS
FIELD	GROUP	SUB-GROUP	Polyline approximation, Waveform analysis, Data compression, Recursive least squares, Generalized likelihood ratio, Kalman filter

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

New algorithms are derived for the polyline approximation of digital waveforms, using recursive least squares and generalized likelihood ratio techniques. Numerical experiments indicate that the new algorithms may in some applications offer significant speed and accuracy advantages compared to prior polyline algorithms.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (*Include Area Code*)	22c. OFFICE SYMBOL
NAVPRO Security Officer	(301) 953 5442	SE

DD FORM 1473, 84 MAR

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

JHU/APL
TG 1370
JUNE 1988

Technical Memorandum

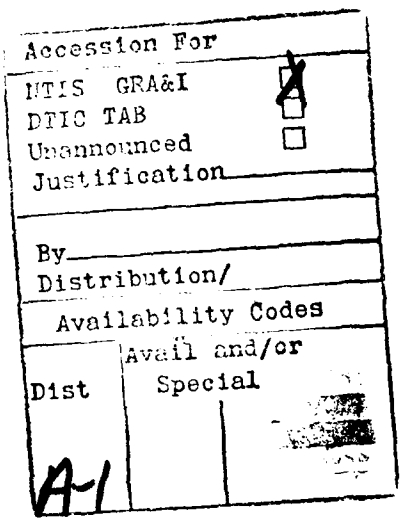
A GLR ALGORITHM FOR POLYLINE APPROXIMATION OF WAVEFORMS

R. A. STEINBERG

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20707
Operating under Contract N00039-87-C-5301 with the Department of the Navy

ABSTRACT

New algorithms are derived for polyline approximation of digital waveforms, using recursive least squares and generalized likelihood ratio techniques. Numerical experiments indicate that the new algorithms may in some applications offer significant speed and accuracy advantages compared to prior polyline algorithms.



CONTENTS

List of Figures 6

List of Tables 7

1.0 Introduction 9

2.0 Detecting Changes in Linear Systems 12
2.1 Problem Formulation 12
2.2 Kalman Preprocessor 13
2.3 GLR Formalism 14

3.0 State Variable Representation of Noise-Free Polylines 18

4.0 Recursive Least-Squares Algorithm 20
4.1 One-Parameter Recursive Regression 20
4.2 RLS Polyline Algorithms 23
4.3 Reformulated State Equations 25

5.0 GLR Polyline Algorithms 26
5.1 Summary of Previous Results 26
5.2 The Auxiliary Variable 27
5.3 Jump Signatures 28
5.4 State Estimate and Innovation Jump Responses 29
5.5 Inverse Covariance of Innovation Estimate 30
5.6 The Filtered Innovation 33
5.7 Generalized Likelihood Ratio 35

6.0 Numerical Experience 36
6.1 Introduction 36
6.2 A Note on Error Metrics and Performance Comparisons 38
6.3 An Experiment at 7:1 Data Compaction 39
6.4 A Closer Look at the GLR Approximation 40
6.5 A Heuristic Modification to GLR 41
6.6 Robustness with Respect to Data Compaction 44

7.0 Conclusions 47

Acknowledgments 48

References 49

Bibliography 50

Appendixes
A. Willsky and Jones GLR Formalism 53
B. One-Parameter Recursive Regression 58
C. Solution for the Auxiliary Variable 60
D. Numerical Experiments at 7:1 Data Compaction 62
E. Performance Statistics versus Data Compaction 69

FIGURES

1. Kalman filter flow of operations....14
2. GLR filter flow of operations. (a) Full GLR. (b) Windowed GLR (M = 20)....17
3. (a) Noise-free, two-segment polyline; (b) slope; (c) offset....19
4. Structure of RLS polyline algorithms....23
5. Application of Eqs. 70 and 71 to Fig. 3a. The polyline shown in Fig. 3a has been translated such that the end of the first line segment is at the starting point of the second line segment....24
6. Numerical experiment validating the closed-form expression for innovation jump response, Eq. 101. (a) Two-segment polyline with parameters (α_0,α,r) = (0.1,0.8,250) (cf. Fig. 3a). (b) Innovation jump response (solid line). The difference between the analytical expression, Eq. 101, and the innovation generated by recursive regression, Eq. 61, is imperceptible on the scale of the plot....31
7. Numerical experiment validating the closed-form expression for innovation jump response, Eq. 101. (a) Two-segment polyline with parameters (α_0,α,r) = (1.5, −1.75,100) (cf. Fig. 3a). (b) Innovation jump reponse (solid line). The difference between the analytical expression, Eq. 101, and the innovation generated by recursive regression, Eq. 61, is imperceptible on the scale of the plot....32
8. (a) Infrared cloud/sky waveform used in testing polyline algorithms. (b) Polyline approximation composed of 49 segments, generated by algorithm GLR-M....37
9. GLR polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (+). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve)....42
10. Data in the neighborhoods of the three points of worst fit for the GLR algorithm. Figure 9a shows these neighborhoods within their larger context; Table 4 provides these data in tabular form. (a) Neighborhood of peak error point. (b) Neighborhood of second worst point. (c) Neighborhood of third worst point....43
11. GLR-M polyline algorithm evaluation. (a) IR cloud waveform (solid curve) and polyline knots (o). (b) Pointwise fitting error. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve)....44
12. Performance statistics as functions of data compaction for GLR (solid curves) and CIM (dashed curves)....45
13. Performance statistics as functions of data compaction for RLS1 (solid curves) and CIM (dashed curves)....46
D-1 EMK and CIM polyline algorithm evaluations. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve)....63

D-2 DP polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).....64
D-3 LSBPT polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).....65
D-4 HOP-S polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).....66
D-5 BAD polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).....67
D-6 RLS1 and RLS2 polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (o). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).....68
E-1 Performance statistics as functions of data compaction for EMK (solid curves) and CIM (dashed curves).....70
E-2 Performance statistics as functions of data compaction for DP (solid curves) and CIM (dashed curves).....71
E-3 Performance statistics as functions of data compaction for LSBPT (solid curves) and CIM (dashed curves).....72
E-4 Performance statistics as functions of data compaction for HOP-S (solid curves) and CIM (dashed curves).....73
E-5 Performance statistics as functions of data compaction for BAD (solid curves) and CIM (dashed curves).....74
E-6 Performance statistics as functions of data compaction for RLS2 (solid curves) and CIM (dashed curves).....75
E-7 Performance statistics as functions of data compaction for RLS1 (solid curves) and CIM (dashed curves).....76

TABLES

1. Application of GLR to polyline approximation results in the development of closed-form expressions for these quantities.....18
2. Polyline algorithms evaluated numerically.....38
3. Performance of polyline algorithms in developing 50-knot approximations to an infrared cloud/sky waveform, Fig. 8a.....39
4. Data in the neighborhoods of the three points of worst fit for the GLR algorithm.....40
5. Performance of three polyline algorithms.....43

1.0 INTRODUCTION

The extraction of time-domain information from a waveform (e.g., pulse shape) can be facilitated by preprocessing the waveform with a "polyline algorithm," i.e., an algorithm that approximates the waveform as a concatenated sequence of straight-line segments. The polyline approximation, if properly accomplished, smooths inconsequential noise structures while leaving significant structure well defined and unsmoothed. Moreover, the polyline representation generally provides a large degree of data compaction relative to the original time series.

Polyline approximation has recently been applied to a great diversity of problem areas (see the Bibliography).

An early polyline application that in many ways anticipated our present work is provided by McAulay and Denlinger[1] in their development of decision-directed tracking algorithms. An early scan-along algorithm was developed by Tomek, based on the notion of finding the longest approximating line segment that can be confined completely within a pair of parallel lines separated by a prespecified error tolerance.[2,3] Some of the difficulties experienced with Tomek's algorithm are alleviated by a scan-along algorithm variously referred to as the cone intersection and minimum perimeter polygon method, discovered independently by Williams[4] and by Sklansky and Gonzalez.[5]

Pavlidis[3] and Pavlidis and Horowitz[6] describe a split-and-merge technique that progressively improves on an initial segmentation until an a priori error specification is satisfied. More recently, Pavlidis[7] has devised an algorithm that recasts his earlier split-and-merge approach into a scan-along structure that he refers to as a hop-along algorithm.

The scan-along and split-and-merge algorithms, while effective in applications, are not intended to be optimal in any sense. Vandewalle[8] has provided an algorithm that, while slow in execution, is intended to provide an approximation that is optimal, in the sense of requiring the minimum number of breakpoints (or knots) to achieve a prespecified error norm. Similar concepts are cast into a somewhat more formal setting by McLaughlin and Zacharski,[9] who refer to their algorithm as the method of E-maximal knots.

A dynamic programming algorithm for polyline approximation has recently been devised independently by Papakonstantinou[10] and by Dunham.[11] These authors

---

[1] R. J. McAulay and E. J. Denlinger, "A Decision-Directed Adaptive Tracker," *IEEE Trans. Aerosp. Electron. Syst.* **AES-9**, 229–236 (1973).

[2] I. Tomek, "Two Algorithms for Piecewise Linear Continuous Approximation of Functions of One Variable," *IEEE Trans. Comput.* **C-23**, 445–448 (1974).

[3] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin (1977).

[4] C. M. Williams, "An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves," *Comput. Graph. Image Process.* **8**, 286–293 (1978).

[5] J. Sklansky and V. Gonzalez, "Fast Polygonal Approximation of Digitized Curves," in *Proc. 1979 IEEE Computer Society Conf. Pattern Recognition Image Processing*, pp. 604–609 (1979).

[6] T. Pavlidis and S. L. Horowitz, "Segmentation of Plane Curves," *IEEE Trans. Comput.* **C-23**, 860–870 (1974).

[7] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Md. (1982).

[8] J. Vandewalle, "On the Calculation of the Piecewise Linear Approximation to a Discrete Function," *IEEE Trans. Comput.* **C-24**, 843–846 (1975).

[9] H. W. McLaughlin and J. J. Zacharski, "Segmented Approximation," in *Approximation Theory*, E. W. Cheney, ed., Academic Press, New York, pp. 647–654 (1980).

[10] G. Papakonstantinou, "Optimal Polygonal Approximation of Digital Curves," *Signal Process.* **8**, 131–135 (1985).

[11] J. G. Dunham, "Optimum Uniform Piecewise Linear Approximation of Planar Curves," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 67–75 (1986).

claim optimality for their algorithms, in the sense of obtaining an approximation with a globally minimum number of knots, for a prespecified peak fitting error. The dynamic programming algorithms[10,11] call upon the cone intersection method as a subroutine.

Numerical experience with the "optimal" algorithms indicates that the dynamic programming algorithm provides both reduced fitting error and faster execution speed, relative to the method of E-maximal knots, as discussed in Section 6 below.

In addition to providing a new dynamic programming algorithm, Dunham presents performance comparisons of several prior polyline algorithms against three test contours.[11] The measures of algorithm performance used by Dunham are execution speed, degree of data compaction, and peak absolute fitting error; algorithms evaluated included those by Pavlidis and Horowitz,[7] Pavlidis,[7] Williams,[4] Sklansky and Gonzalez,[5] Badi'i and Peikari,[12] Ramer,[13] Roberge,[14] and Dunham.[11] Dunham's results show that:

1. Roberge's algorithm, while consistently fastest in execution, typically achieved only half the data compaction ratio of the optimal dynamic programming solution (for fixed peak error);
2. The Sklansky/Gonzalez algorithm executed about as fast as the fastest of the other algorithms (not including Roberge's) and consistently achieved data compaction ratios almost as good as the dynamic programming algorithm.

Consequently, the Sklansky/Gonzalez algorithm may be considered a benchmark against which to compare the performance of other methods.

Pavlidis, in editorial commentary,[15] has noted that performance comparisons such as Dunham's are clouded by uncertainties in algorithm implementation. However, also as noted by Pavlidis, such numerical studies and algorithm intercomparisons are nonetheless valuable. Following Dunham, we presently both introduce new algorithms and compare them to prior ones. Comparing our work with Dunham's:

1. Our interest is in approximating waveforms rather than two-dimensional contours.
2. We have expanded the scope of error metrics to include average error (i.e., bias), root mean square error, and average absolute error.
3. Our selection of polyline algorithms for evaluation and comparison, partially overlapping Dunham's, includes those of Pavlidis,[7] Papakonstantinou,[10] Sklansky and Gonzalez,[5] Wall and Danielsson,[16] Tomek,[2] and McLaughlin and Zacharski,[9] as well as the new algorithms presented in this report.

As discussed in Section 6, the relative performance of the various algorithms turns out to be somewhat dependent on the error metric. In particular, the original algorithms presented here are considerably superior to prior algorithms, with respect to bias and room mean square error.

Our new algorithms are of two varieties, which we refer to as recursive least-squares (RLS) and generalized likelihood ratio (GLR) algorithms.

---

[12]F. Badi'i and B. Peikari, "Functional Approximation of Planar Curves via Adaptive Segmentation," *Int. J. Systems Sci.* **13**, 667–674 (1982).

[13]U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Comput. Graph. Image Process.* **1**, 244–256 (1972).

[14]J. Roberge, "A Data Reduction Algorithm for Planar Curves," *Comput. Vision Graph. Image Process.* **29**, 168–195 (1985).

[15]T. Pavlidis, "Editorial—Papers on Shape Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1 (1986).

[16]K. Wall and P. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves," *Comput. Graph. Image Process.* **28**, 220–227 (1984).

Our RLS algorithms are derived in Section 4 based on a simple Kalman filter. Our numerical experience with these algorithms is that, relative to the Sklansky/Gonzalez algorithm, RLS has

- Faster execution speed
- Slightly worse peak error
- Less average error
- Smaller room mean square error.

The original application of the statistical GLR method to formulating the problem of change detection in linear systems was provided by McAulay and Denlinger.[1] A general solution to the GLR formulation was provided by Willsky and Jones.[17] In Section 2 of this report we review the Willsky/Jones GLR formalism that, in Section 5, we particularize to the problem of polyline approximation.

The most direct antecedent of our work is McAulay and Denlinger.[1] Features common to our work and theirs are:

1. Use of a GLR formulation for change detection;
2. "...A piecewise linear model... to [describe] the noiseless evolution..." of measured waveforms;[1]
3. Tracking of "...straight-line segments with a simple two-term Kalman filter...".[1]

Some points of distinction between our approach and theirs are:

1. We develop a one-parameter recursive regression as our candidate no-jump Kalman filter. The corresponding filter used in Ref. 1 is unknown since its discussion is quite brief.
2. Our "jump signature" solution in Section 5.3 is exact, derived as a special case of the general solution in Ref. 17. The corresponding signature presented in Ref. 1 is approximate, being based on numerical experience (cf. the discussion in connection with Eq. 106).

Although Ref. 1 is widely cited as the original GLR change detection formulation, the casting of target tracking problems in terms of an equivalent polyline approximation has apparently gone unnoticed in the literature, perhaps due to the brevity of this part of the discussion in Ref. 1. Nevertheless, we see this as potentially of high interest, since algorithms for polyline approximation can be applied to target tracking and vice versa. While there has very recently been some renewed interest in applying polyline approximation methodology to problems in radar tracking,[18] there may still be considerable unexploited potential for cross-fertilization between the two areas.

[17] A. S. Willsky and H. L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," *IEEE Trans. Auto. Control* **AC-21**, 108–112 (1976).
[18] S. F. Haase, "Advanced Radar Tracking Techniques," in *IR&D/B&P Program Plan Vol. II FY1988*, JHU APL, Laurel, Md., pp. II-375 to II-397 (1988).

2.0 DETECTING CHANGES IN LINEAR SYSTEMS

2.1 PROBLEM FORMULATION

This section follows Willsky and Jones[17] and like them provides results without derivations. For completeness, the appropriate derivations are provided in Appendix A.

We assume a discrete-time system whose dynamics evolve according to the state and measurement equations:

$$x(k+1) = \Phi(k+1,k) \cdot x(k) + \Gamma(k) \cdot w(k) + \delta_{\theta,k+1} \cdot \nu \ , \qquad (1)$$

$$z(k+1) = H(k+1) \cdot x(k+1) + v(k+1) \ , \qquad (2)$$

with all quantities being real valued (see the list below for definitions). Quantities x, w, ν, z, and v may be vectors, while Φ, Γ, and H are real matrices of commensurate dimensionality.

Quantity	Definition
k	Discrete time
x	State vector
w	State noise
z	Measurement/observation
v	Measurement noise
δ	Dirac delta function
ν	Jump amplitude
θ	Jump occurrence time
Φ	State transition matrix
H	Measurement matrix

The noise sequences $w(k)$ and $v(k)$ are assumed to be zero-mean and Gaussian, with covariances

$$\text{cov}[w(k)] = E[w(k)w'(k)] = Q(k) \ , \qquad (3)$$

$$\text{cov}[v(k)] = R(k) \ , \qquad (4)$$

where in Eq. 3 $E[\cdot]$ denotes ensemble expectation, and a primed quantity denotes the vector transpose.

The Dirac delta term in Eq. 1 indicates the presence of a discontinuity (or "jump") in what would otherwise be a smooth temporal evolution in the state vector. The jump amplitude ν and time of occurrence θ are presumed to be unknown, subsuming the possibility that a jump never occurs ($\nu = 0$ or $\theta = \infty$).

The objective of jump detection processing is to determine whether a jump has occurred and, if so, to establish accurate estimates for ν and θ by means of processing operations performed on the measurements $z(k)$.

Since Eq. 1 has the stochastic driving term w, the state x must necessarily be a random process.

2.2 KALMAN PREPROCESSOR

Our approach to detecting the presence of a jump in Eq. 1 is to process the data with a filter that will be optimal if no jump occurs and to monitor the filter's performance to assess the continuing validity of the no-jump hypothesis. In the control systems literature, this approach is sometimes referred to as analytical redundancy, and the filter is sometimes called a no-fail observer or normal-mode observer.[19]

Subject to the no-jump hypothesis (i.e., prior to the jump, when $k < \theta$), the optimal approach for estimating the state is provided by a Kalman-Bucy filter:

$$\hat{x}(0|0), \text{ initial state estimate} \tag{5}$$

$$P(0|0), \text{ initial covariance} \tag{6}$$

$$\hat{x}(k+1|k) = \Phi(k+1,k) \cdot \hat{x}(k|k) \tag{7}$$

$$\hat{z}(k+1|k) = H(k+1) \cdot \hat{x}(k+1|k) \tag{8}$$

$$\gamma(k+1) = z(k+1) - \hat{z}(k+1|k) \tag{9}$$

$$P(k+1|k) = \Phi(k+1,k) \cdot P(k|k) \cdot \Phi'(k+1,k) + \Gamma(k) \cdot Q(k) \cdot \Gamma'(k) \tag{10}$$

$$V(k+1) = H(k+1) \cdot P(k+1|k) \cdot H'(k+1) + R(k+1) \tag{11}$$

$$K(k+1) = P(k+1|k) \cdot H'(k+1) \cdot V^{-1}(k+1) \tag{12}$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1) \cdot \gamma(k+1) \tag{13}$$

$$P(k+1|k+1) = [I - K(k+1) \cdot H(k+1)] \cdot P(k+1|k) \tag{14}$$

with all quantities in Eqs. 5–14 being real valued. (See the list below for definitions; note that quantity Z^j denotes the sequence of observations $z(1), z(2), \ldots, z(j)$.) The flow of operations in the Kalman filter is depicted in Fig. 1.

Quantity	Definition
$\hat{x}(k\|j)$	Conditional mean of $x(k)$, given Z^j
$\hat{z}(k\|j)$	Conditional mean of $z(k)$, given Z^j
$\gamma(k)$	Innovation
$P(k\|j)$	Covariance of $x(k\|j)$
$V(k)$	Covariance of $\gamma(k)$
$K(k)$	Kalman gain

[19] A. Madiwale and B. Friedland, "Comaprison of Innovations-Based Analytical Redundancy Methods," in *Proc. 1983 Am. Control Conf.*, pp. 940–945 (1983).

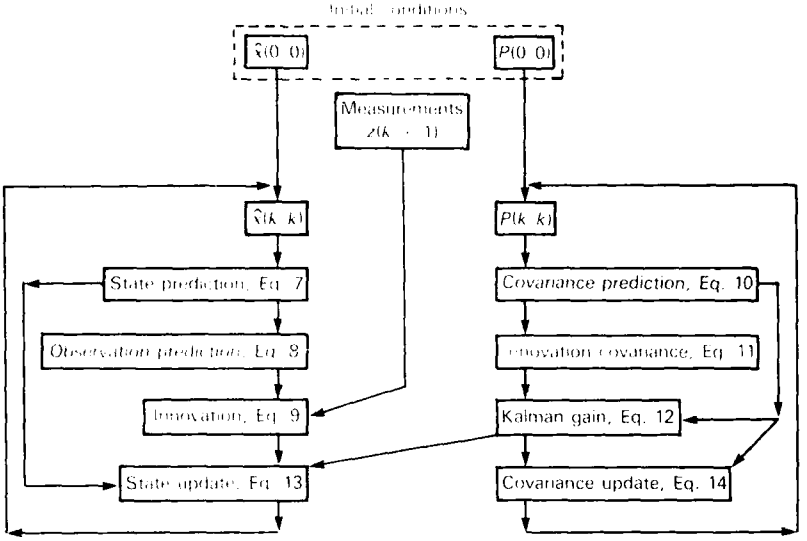


Figure 1 Kalman filter flow of operations.

2.3 GLR FORMALISM

As first noted by McAulay and Denlinger,[1] the linearity of the system equations allows their solution in response to a jump excitation to be written as the sum of a jump-independent stochastic component (x_1) and a second component (x_2) linearly proportional to the jump amplitude. Moreover, with the jump amplitude and time taken as deterministic (though unknown) quantities, the jump-dependent component x_2 is seen to be deterministic.

Following Ref. 17, we write the state as

$$x(k) = x_1(k) + \Phi(k,\theta) \cdot \nu \ , \tag{15}$$

where $x_1(k)$ is the value that $x(k)$ would take in the absence of a jump ($\nu = 0$), and $\Phi(k,\theta) \cdot \nu$ is the state perturbation in response to a jump, where

$$\Phi(k,\theta) = 0, \qquad k < \theta \tag{16}$$

$$\Phi(\theta,\theta) = I \tag{17}$$

$$\Phi(k+1,\theta) = \Phi(k+1,k) \cdot \Phi(k,\theta) \ . \tag{18}$$

The proof of Eqs. 16–18 is provided in Appendix A, along with the proofs for other results presented in this section.

From Eqs. 2 and 15,

$$z(k) = z_1(k) + H(k) \cdot \Phi(k,\theta) \cdot \nu \ . \tag{19}$$

Similarly, the linearity of the Kalman filter equations allows their solution to be written in a form analogous to Eqs. 15 and 19, viz., for the state estimate,

$$\hat{x}(k|k) = \hat{x}_1(k|k) + F(k;\theta) \cdot \nu \ , \tag{20}$$

and for the innovation,

$$\gamma(k) = \gamma_1(k) + G(k;\theta) \cdot \nu \ . \tag{21}$$

The quantities $F(k;\theta)$ and $G(k;\theta)$ appearing in Eqs. 20 and 21 are sometimes referred to as the jump "signatures"; they characterize the Kalman filter's response to a jump excitation in the system equations.

It clearly must be true that

$$\left.\begin{aligned} F(k;\theta) &= 0 \\ G(k;\theta) &= 0 \end{aligned}\right\} \qquad k < \theta \tag{22}$$

in order to preserve the definitions of $\hat{x}_1$ and γ_1 as the Kalman filter responses in the absence of a jump.

It can be shown that $G(k;\theta)$ may be written in terms of $F(k;\theta)$:

$$G(k;\theta) = H(k) \cdot [\Phi(k,\theta) - \Phi(k,k-1) \cdot F(k-1;\theta)] \ . \tag{23}$$

The function $F(k;\theta)$ appearing in Eqs. 20 and 23 is given by

$$F(k;\theta) = \sum_{j=\theta}^{k} \Theta(k;j) \cdot K(j) \cdot H(j) \cdot \Phi(j,\theta) \ , \tag{24}$$

where the auxiliary variable $\Theta(k;j)$ is obtained recursively as

$$\Theta(k;\theta) = [I - K(k) \cdot H(k)] \cdot \Phi(k,k-1) \cdot \Theta(k-1,\theta) \tag{25}$$

$$\Theta(k;\theta) = 0, \qquad k < \theta \tag{26}$$

$$\Theta(\theta;\theta) = I \tag{27}$$

Assuming that the measurements $z(k)$ generated according to Eqs. 1 and 2 are processed by the Kalman filter, Eqs. 5-14, the innovation $\gamma(k)$ will be a zero-mean Gaussian process (γ_1) until the jump occurs at $k = \theta$, whereupon $\gamma(k)$ will develop a bias, $G(k;\theta) \cdot \nu$. Jump detection is accomplished by detecting the innovation bias, which, in turn, is accomplished by a form of matched filtering applied to the innovation process.

Writing the GLR estimates for ν and θ as $\hat{\nu}$ and $\hat{\theta}$, it can be shown that

$$\hat{\nu}(k) = C^{-1}(k;\theta) \cdot d(k;\theta)|_{\theta=\hat{\theta}(k)} \ , \tag{28}$$

where $d(k;\theta)$ is obtained via a matched linear filtering operation on the innovation process,

$$d(k;\theta) = \sum_{n=\theta}^{k} G'(n;\theta) \cdot V^{-1}(n) \cdot \gamma(n) \ , \tag{29}$$

and matrix $C^{-1}(k;\theta)$ is the covariance of $\hat{\nu}$, computed as

$$C(k;\theta) = \sum_{n=\theta}^{k} G'(n;\theta) \cdot V^{-1}(n) \cdot G(n;\theta) \ . \tag{30}$$

The GLR estimate for jump time, $\hat{\theta}$, is obtained as the value of θ that maximizes the generalized likelihood ratio, $\ell(k;\theta)$, i.e.,

$$\ell(k;\hat{\theta}) = \max_{\theta = 1,2,\ldots,k} \ell(k;\theta) , \tag{31}$$

where

$$\ell(k;\theta) = d'(k;\theta) \cdot C^{-1}(k;\theta) \cdot d(k;\theta) . \tag{32}$$

The GLR algorithm determines that a jump has occurred whenever $\ell(k;\hat{\theta})$ exceeds a fixed threshold, ϵ; the GLR estimate for the jump occurrence time is then provided by Eq. 31, and the GLR estimate for jump amplitude $\hat{\nu}$ is given explicitly by Eq. 28.

We note that the function $C(k;\theta)$ given by Eq. 30 is computable off line, as is the function $[G'(j;\theta) \cdot V^{-1}(j)]$ in Eq. 29.

The GLR filter flow of operations is diagrammed in Fig. 2a. Maximization in Eq. 31 occurs over k values of θ, corresponding to the k parallel branches in Fig. 2a. Because the number of calculations required to implement Eq. 31 increases with time, implementing "full GLR" requires progressively increasing numbers of calculations at successive time steps—an undesirable property. An approach to bounding the number of required calculations is to restrict the range of θ values in Eq. 31 to the M "most recent" values, i.e., to the range $[(k - M + 1),k]$ (Fig. 2b).

When the jump vector ν can be written as

$$\nu = \alpha \cdot f(\theta) , \tag{33}$$

where α is an unknown scalar and $f(\theta)$ is a known vector function of θ, the formulations for $\hat{\nu}$ and $\ell(k;\theta)$ may be written in the following forms, alternatives to Eqs. 28 and 32:

$$\hat{\nu}(k) = \hat{\alpha}(k;\hat{\theta}) \cdot f(\hat{\theta}) \tag{34}$$

$$\hat{\alpha}(k;\theta) = b(k;\theta)/a(k;\theta) \tag{35}$$

$$\ell(k;\theta) = b^2(k;\theta)/a(k;\theta) \tag{36}$$

where

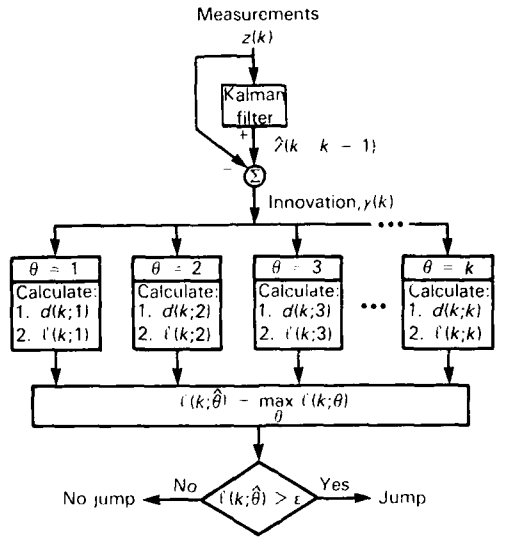
$$a(k;\theta) = f'(\theta) \cdot C(k;\theta) \cdot f(\theta) , \tag{37a}$$

and

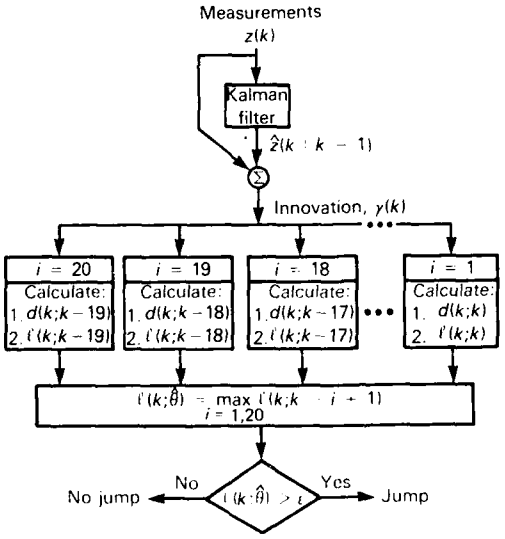
$$b(k;\theta) = f'(\theta) \cdot d(k;\theta) . \tag{37b}$$

We note that Basseville and Benveniste[20] have suggested a modified form of the Willsky/Jones algorithm in which thresholding is performed on the estimated jump amplitude $\hat{\nu}$, given by Eq. 28. Much simpler still, in Section 4.2 we propose thresholding the normalized innovation $\gamma(k)/(V(k))^{1/2}$ directly, i.e., using the Kalman filter alone, with no additional processing other than thresholding, for jump detection.

[20]M. Basseville and A. Benveniste, "Design and Comparative Study of Some Sequential Jump Detection Algorithms for Digital Signals," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-31**, 521-535 (1983).



(a) Full GLR.



(b) Windowed GLR ($M = 20$).

Figure 2 GLR filter flow of operations.

The GLR algorithm is not altogether simple to understand. As perhaps the simplest application yet put forward of the GLR method, our GLR polyline development in Section 5 helps cast insight into the structure of GLR generally. Some of the simplifying features of our development are:

1. Low dimensionality: 2×1 vectors and 2×2 matrices are our highest-dimensional entities.
2. Closed forms: A number of quantities, generally available only as the solution of recursion relations, are obtained in closed form (see Table 1).

Table 1
Application of GLR to polyline approximation results in the development of closed-form expressions for these quantities.

Variable	Definition	Open Form	Closed Form
$K(k)$	Kalman gain	Eq. 12	Eq. 62
$V(k)$	Innovation covariance	Eq. 11	Eq. 63
$\Theta(k;\theta)$	Auxiliary variable	Eqs. 25–27	Eqs. 87 and 89
$F(k;\theta)$	State signature	Eq. 24	Eq. 95
$G(k;\theta)$	Innovation signature	Eq. 23	Eq. 97
$C(k;\theta)$	$\hat{\nu}$ inverse covariance	Eq. 30	Eq. 109

3.0 STATE VARIABLE REPRESENTATION OF NOISE-FREE POLYLINES

In this section we show that the polyline waveform approximation problem can be given a state variable formulation, as required for application of the GLR formalism.

Underlying our development is the assumption that waveforms of interest may be approximated as noise-free polylines, with additive, zero-mean, stationary Gaussian noise. Thus, our interest in this section is to cast the equations of a noise-free polyline into the form of Eqs. 1 and 2, in which the noise terms have been set to zero:

$$x(k+1) = \Phi(k+1,k) \cdot x(k) + \delta_{n,k+1} \cdot \nu \ , \tag{38}$$

$$z(k+1) = H(k+1) \cdot x(k+1) \ . \tag{39}$$

We assume that the ordinate of the initial breakpoint is known; subtracting the initial ordinate then results in a polyline whose initial breakpoint is at coordinates $(k_0,0)$. For further discussion, see Fig. 3, where we define

$$\begin{aligned} j &\equiv k \quad k_0 \ , \\ r &\equiv 0 \quad k_0 \ . \end{aligned} \tag{40}$$

Consideration of Fig. 3 shows that we may write for the polyline slope

$$s(j) \quad \alpha_0 \cdot \alpha \cdot \mu(j \quad r) \ . \tag{41}$$

where $\mu(\cdot)$ is the unit-step function. For offset we may write

$$y(j) \quad \alpha \cdot r \cdot \mu(j \quad r) \ . \tag{42}$$

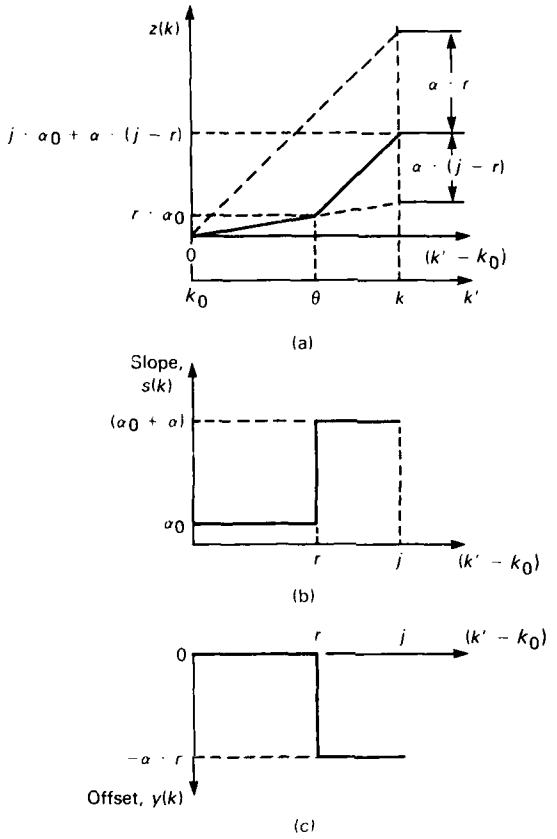


Figure 3 (a) Noise-free, two-segment polyline; (b) slope; (c) offset.

Also, we note that

$$\mu(j - r + 1) - \mu(j - r) = \delta_{j+1,r} \ , \tag{43}$$

where δ is the Dirac delta.

From the foregoing, we obtain the desired state-variable representation for the noise-free polyline:

$$\begin{Bmatrix} y(j+1) \\ s(j+1) \end{Bmatrix} = \begin{Bmatrix} y(j) \\ s(j) \end{Bmatrix} + \delta_{r,j+1} \cdot \alpha \cdot \begin{Bmatrix} -r \\ 1 \end{Bmatrix} \ , \tag{44}$$

and

$$z(k_0 + j) = \{ \ 1 \quad j \ \} \cdot \begin{Bmatrix} y(j) \\ s(j) \end{Bmatrix} \ . \tag{45}$$

Comparing Eqs. 44 and 45 with Eqs. 38 and 39, we obtain

$$x(k_0 + j) = \begin{Bmatrix} y(j) \\ s(j) \end{Bmatrix} \tag{46}$$

$$\Phi(k+1,k) = I \tag{47}$$

$$v = \alpha \cdot \begin{Bmatrix} -r \\ 1 \end{Bmatrix} \tag{48}$$

$$H(k_0 + j) = \begin{Bmatrix} 1 & j \end{Bmatrix} . \tag{49}$$

Comparing Eqs. 48 and 33, we see that they are of the same form, permitting us to write

$$f(k_0 + r) = \begin{Bmatrix} -(\theta - k_0) \\ 1 \end{Bmatrix} = \begin{Bmatrix} -r \\ 1 \end{Bmatrix} . \tag{50}$$

4.0 RECURSIVE LEAST-SQUARES ALGORITHMS

This section comprises three subsections. In 4.1 we derive a recursive least-squares regression formula for fitting a straight line to a set of random data, as a special case of the Kalman filter. We assume that the initial coordinates of the fitted line are known and only the line slope must be estimated. Since only a single parameter (slope) is being estimated, our resulting "one-parameter recursive regression" is somewhat different from the usual two-parameter intercept/slope regression formula.[21] In 4.2 we use our one-parameter regression formula to derive two algorithms for polyline approximation, which we subsequently refer to as RLS algorithms. The performance of these algorithms is compared in Section 6 to other algorithms. In 4.3 we slightly recast our one-parameter regression results into a form suitable for use with the general GLR formulation presented in Section 2.

4.1 ONE-PARAMETER RECURSIVE REGRESSION

Taking $j < r$ in Eqs. 41 and 42, we obtain the equations of a straight line, viz.,

$$y(j) = 0 \tag{51a}$$

$$s(j) = \alpha_0 \tag{51b}$$

According to Eqs. 51, the line passes through initial coordinates $(k_0,0)$, with slope α_0.

From Eqs. 45 and 51a, and adding a noise term (as in Eq. 2), we obtain

[21] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press (1988).

$$z(k) = \begin{cases} 0 & , \quad j = (k - k_0) = 0 \\ j \cdot s(j) + v(k) & , \quad 0 < j = (k - k_0) < r \end{cases} , \tag{52}$$

where we assume that $v(k)$ is zero mean, Gaussian, and stationary:

$$R(k) = \text{cov}[v(k)] = \sigma^2 . \tag{53}$$

Under the assumption that the initial ordinate of the line is known and can be subtracted to obtain a line that goes through $(k_0,0)$ (cf. Fig. 3), the single unknown parameter remaining is simply the line's slope, $s(j) = \alpha_0$. Although our constraint, $z(k_0) = 0$, may seem artificial, it is justified later in connection with our application of the results in this section.

A simple recursive procedure for estimating the parameter $s(j) = \alpha_0$ in Eq. 52 may be developed from the Kalman filter Eqs. 5–14, in which all generally vector and matrix quantities reduce presently to scalars. We identify the line slope $s(j)$ as the "system state," which is time-independent (Eq. 51b). Consequently,

$$\hat{s}(n|j) = \hat{s}(m|j) , \tag{54}$$

$$P_s(n|j) = P_s(m|j) , \tag{55}$$

for all values of n, m, and j, and where we write P_s as the covariance of $\hat{s}$. It follows that we may simplify notation by writing

$$\begin{aligned} \hat{s}(j) &\equiv \hat{s}(n|j) , \\ P_s(j) &\equiv P_s(n|j) . \end{aligned} \tag{56}$$

Equation 51b may be written in the form of Eq. 1 by making the identification

$$\Phi(k + 1,k) = 1 \tag{57}$$

$$w(k) = 0 . \tag{58}$$

From Eqs. 58 and 3, $Q(k) = 0$. Similarly, Eq. 52 can be cast in the form of Eq. 2 by writing

$$H(k) = (K - k_0) = j . \tag{59}$$

Using Eqs. 53 and 56–59, we obtain, corresponding to the Kalman filter Eqs. 8–14,

$$\hat{z}(k - 1) = j \cdot \hat{s}(j - 1) \tag{8'}$$

$$\gamma(k) = z(k) - \hat{z}(k - 1) \tag{9'}$$

$$P_s(j) = P_s(j) \tag{10'}$$

$$V(k) = j^2 \cdot P_s(j - 1) + \sigma^2 \tag{11'}$$

$$K_s(j) = j \cdot P_s(j - 1)/V(k) \tag{12'}$$

$$\hat{s}(j) = \hat{s}(j - 1) + K_s(j) \cdot \gamma(k) \tag{13'}$$

$$P_s(j) = [1 - j \cdot K_s(j)] \cdot P_s(j - 1) , \tag{14'}$$

where it is understood that $k \equiv (k_0 + j)$ in Eqs. 8′-14′.

As shown in Appendix B, the desired recursive estimation procedure for line slope is obtained from Eqs. 8′-14′ as

$$\hat{s}(j) = \hat{s}(j-1) + K_s(j) \cdot \gamma(k) , \qquad j = (k - k_0) \geq 2 , \tag{60}$$

where

$$\gamma(k) = z(k) - j \cdot \hat{s}(j-1) , \tag{61}$$

and

$$K_s(j) = \frac{6}{(j+1) \cdot (2j+1)} . \tag{62}$$

We also find in Appendix B that

$$V(k) \equiv V_0(j) = \sigma^2 \cdot \left(\frac{j+1}{j-1}\right) \cdot \left(\frac{2j+1}{2j-1}\right) , \tag{63}$$

and

$$P_s(j) = \frac{6\sigma^2}{j \cdot (j+1) \cdot (2j+1)} . \tag{64}$$

It can be shown that Eqs. 60–64 are equivalent to a recursive least-squares regression for line slope, derived on the assumption that the initial ordinate is specified. The recursion is initialized by taking

$$\hat{s}(1) = z(k_0 + 1) - z(k_0) . \tag{65}$$

The first loop of recursive continuation follows from Eqs. 65, 60, and 61, in the following sequence:

$$\gamma(k_0 + 2) = z(k_0 + 2) - 2 \cdot \hat{s}(1) , \tag{66}$$

$$\hat{s}(2) = \hat{s}(1) + K_s(2) \cdot \gamma(k_0 + 2) . \tag{67}$$

The slope estimate $\hat{s}(1)$ and measurement $z(k_0 + 2)$ are used in Eqs. 66 and 67 to obtain the next slope estimate, $\hat{s}(2)$. More generally, we interpret Eqs. 60 and 61 as a prescription for taking a slope estimate $\hat{s}(j-1)$ and measurement $z(k)$ to derive the next slope estimate $\hat{s}(j)$.

Inspection of Eq. 62 reveals two properties of the Kalman gain that are generally true even for more complex, higher-dimensional Kalman filters, viz.,

1. The Kalman gain, though time-dependent, is independent of the data and may be computed off line.
2. The Kalman gain becomes progessively smaller as increasing amounts of data are processed, i.e.,

$$\lim_{j \to \infty} K_s(j) = 0 . \tag{68}$$

It follows as a consequence of Eqs. 60 and 68 that the RLS slope estimate $\hat{s}(j)$ becomes progressively less sensitive to the measurements as increasing amounts of data are processed. In a sense, the filter becomes increasingly "satisfied" with the goodness of its slope estimate. This may be seen also from Eq. 64: the variance of the estimate, as calculated by the filter, decreases to zero. Naturally, the filter's assessment of its own performance, as provided by Eq. 64, is only valid if the underlying waveform model is correct, i.e., if the data realize a process composed of a straight line and additive, white, Gaussian noise.

Our reasons for assuming that $z(k_0)$ is known, and our basis for choosing a value for $z(k_0)$, are discussed in the next section.

4.2 RLS POLYLINE ALGORITHMS

In this section we describe two polyline approximation algorithms based on the premise that the development of a nonzero trend in $\gamma(k)$ indicates that the underlying model, Eq. 52, is no longer valid and that the measurements can no longer be fitted adequately with a single straight line.

Both of our simple RLS polyline algorithms are structured as follows (Fig. 4).

1. Equations 60 and 61 are applied to the data to generate the innovation sequence $\gamma(k)$.
2. The innovation sequence is tested in some fashion for the development of nonzero bias.
3. When a nonzero innovation bias is detected, say, at time $k = (\theta + 1) = (k_0 + r + 1)$, a breakpoint or "knot" is introduced in the waveform approximation at time $k = \theta$. The ordinate of the knot is estimated as

$$\hat{z}(\theta) = (r + 1) \cdot \hat{s}(r) , \tag{69}$$

which is obtained by substituting $(k - 1) = \theta = (k_0 + r)$ into Eq. 8′, and $\hat{s}(r)$ is the last slope estimate generated by the RLS algorithm.

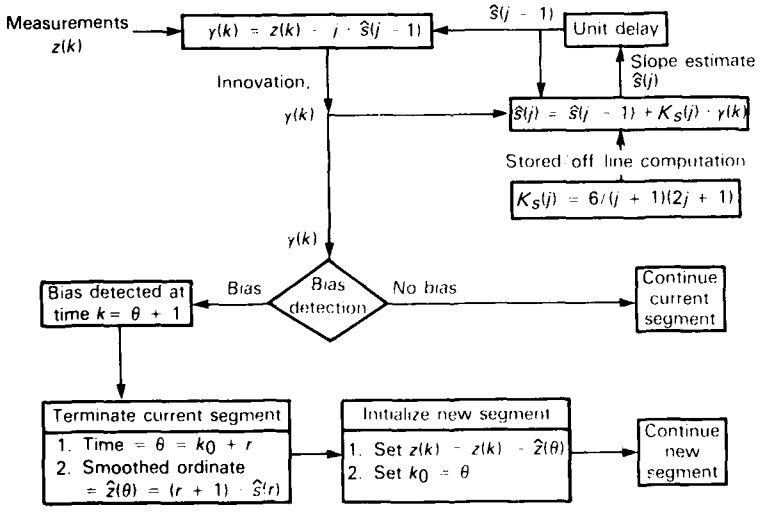


Figure 4 Structure of RLS polyline algorithms.

4. The RLS algorithm is reinitialized by making the assignments

$$z(k) \leftarrow z(k) - \hat{z}(\theta) , \qquad k > \theta \tag{70}$$

$$k_0 \leftarrow \theta . \tag{71}$$

The purpose of Eq. 70 is to translate the polyline such that the origin of the new z axis is at the starting point of the new line segment. Similarly, Eq. 71 translates the time axis such that the starting time of the polyline's new segment is set equal to the ending time of the first segment. Figure 5 illustrates the result of applying Eqs. 70 and 71 to Fig. 3a.

Our two RLS polyline algorithms, RLS1 and RLS2, are distinguished by the means used for detecting innovation bias. In RLS1, we first divide $\gamma(k)$ by the square root of its covariance, given by Eq. 63, to obtain the normalized innovation, $S(k)$:

$$S(k_0 + j) = \gamma(k_0 + j) \cdot \frac{1}{\sigma} \left(\frac{j - 1}{j + 1} \cdot \frac{2j - 1}{2j + 1} \right)^{1/2} . \tag{72}$$

Bias detection is then accomplished by applying a fixed threshold, ϵ, to $S(k)$:

$$\begin{aligned} &\text{If} \qquad |S(\theta + 1)| > \epsilon \\ &\text{Then} \quad \text{terminate current segment at } k = \theta . \end{aligned} \tag{73}$$

As we will discuss later in Section 6, selecting a threshold to provide either a fixed error norm or a specified data compaction ratio requires iteration on the value of ϵ.

Our bias detector RLS2 is composed of two independent criteria such that a bias is declared if either criterion is satisfied. One of the two bias detectors in RLS2 is fixed threshold detection applied to $S(k)$, i.e., Eq. 73. The second bias detector in RLS2 is as follows:

$$\begin{aligned} &\text{If} \qquad \gamma(\theta + n) \cdot \gamma(\theta + n + 1) > 0 , \qquad n = 1,2, \ldots, (N - 1) \\ &\text{Then} \quad \text{terminate current segment at } k = \theta . \end{aligned} \tag{74}$$

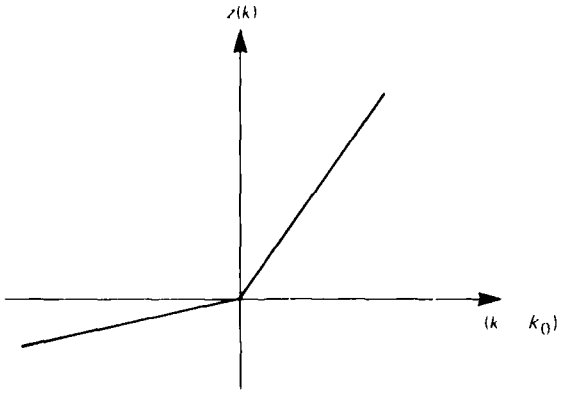


Figure 5 Application of Eqs. 70 and 71 to Fig. 3a. The polyline shown in Fig. 3a has been translated such that the end of the first line segment is at the starting point of the second line segment.

In words, bias is detected as a sequence of N values of $\gamma(k)$ all having the same sign, where N (like ϵ in Eq. 73) is a threshold parameter. The idea behind Eq. 74 is that the innovation sequence should be spectrally white, and therefore subject to frequent sign changes, only so long as the one-segment waveform model (Eq. 52) is valid. A test similar in spirit to Eq. 74 was discussed by Pavlidis (Ref. 7, p. 288, Fig. 12.5).

Our discussion of the performance of the RLS1 and RLS2 polyline algorithms is deferred until Section 6, where their performances are compared to those of a number of other methods.

4.3 REFORMULATED STATE EQUATIONS

In this section we slightly recast our one-parameter regression results into a form suitable for use with the general GLR formalism presented in Section 2.

From Eq. 46,

$$P(k) = \text{cov}[x(k)] = E\left[\begin{Bmatrix} y \\ s \end{Bmatrix} \cdot \{y\ s\}\right] = \begin{Bmatrix} \text{cov}[y(j)] & E[y \cdot s] \\ E[y \cdot s] & \text{cov}[s(j)] \end{Bmatrix} . \tag{75}$$

However, from Eq. 51a,

$$\text{cov}[y] = E[y \cdot s] = 0 , \tag{76}$$

and, from Eq. 64,

$$\text{cov}[s(j)] = P_s(j) = \frac{6\sigma^2}{j \cdot (j+1) \cdot (2j+1)} .$$

From Eqs. 75 and 76,

$$P(k) = P_s(j) \cdot \begin{Bmatrix} 0 & 0 \\ 0 & 1 \end{Bmatrix} . \tag{77}$$

From Eqs. 46 and 56, we simplify notation by writing

$$\hat{x}(k_0 + j) = \begin{Bmatrix} \hat{y}(j) \\ \hat{s}(j) \end{Bmatrix} \equiv \hat{x}(n|k_0 + j) , \tag{78}$$

independent of n.

From Eqs. 12, 49, 77, and 78,

$$K(k) = P_s(j-1) \cdot \begin{Bmatrix} 0 & 0 \\ 0 & 1 \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ j \end{Bmatrix} \cdot V^{-1}(k) ,$$

or

$$K(k) = [j \cdot P_s(j-1)/V(k)] \cdot \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} . \tag{79}$$

From Eqs. 79 and 12′,

$$K(k) = K_s(j) \cdot \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} , \tag{80}$$

where $K_s(j)$ is given by Eq. 62.

5.0 GLR POLYLINE ALGORITHMS

5.1 SUMMARY OF PREVIOUS RESULTS

In this section we gather together for convenient reference the various previously derived quantities needed to particularize the general GLR formalism to the polyline approximation problem.

$$x(k_0 + j) = \begin{Bmatrix} y(j) \\ s(j) \end{Bmatrix} \tag{46}$$

$$\Phi(k+1,k) = I \tag{47}$$

$$r = \alpha \cdot \begin{Bmatrix} r \\ 1 \end{Bmatrix} \tag{48}$$

$$H(k_0 + j) = \{ 1 \quad j \} \tag{49}$$

$$f(k_s + r) = \begin{Bmatrix} (\theta - k_0) \\ 1 \end{Bmatrix} = \begin{Bmatrix} r \\ 1 \end{Bmatrix} \tag{50}$$

$$P(k) = P_s(j) = \begin{Bmatrix} 0 & 0 \\ 0 & 1 \end{Bmatrix} \tag{77}$$

$$\hat{x}(k_0 + j) = \begin{Bmatrix} \hat{y}(j) \\ \hat{s}(j) \end{Bmatrix} = \hat{x}(n' k_0 + j) \tag{78}$$

$$K(k) = K_s(j) = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \tag{80}$$

$$K_s(j) = \frac{6}{(j+1)\cdot(2j+1)} \tag{62}$$

$$V(k) = V_0(j) = \alpha^2 \cdot \binom{j+1}{j-1} \cdot \binom{2j+1}{2j-1} \tag{63}$$

$$P_1(j) = \frac{6\sigma^2}{j \cdot (j+1) \cdot (2j+1)} \tag{64}$$

We recall that, throughout the above equations,

$$\begin{aligned} j &\equiv k - k_0 \ , \\ r &\equiv \theta - k_0 \ . \end{aligned} \tag{40}$$

From Eqs. 16-18 and 47 we have

$$\Phi(k,\theta) = \begin{cases} 0 \ , & k < \theta \\ I \ , & k \geq \theta \end{cases} . \tag{81}$$

5.2 THE AUXILIARY VARIABLE

From Eqs. 49 and 80,

$$K(k) \cdot H(k) = K_1(j) \cdot \begin{Bmatrix} 0 & 0 \\ 1 & j \end{Bmatrix} . \tag{82}$$

It then follows from Eqs. 81 and 82 that the difference equation for the auxiliary variable, Eq. 25, presently simplifies to

$$\Theta(k_0 + j;\theta) = \begin{Bmatrix} 1 & 0 \\ -K_1(j) & [1 - j \cdot K_1(j)] \end{Bmatrix} \cdot \Theta(k_0 + j - 1;\theta) \ . \tag{83}$$

With the definition

$$\Theta(k_0 + j;k_0 + r) = \begin{Bmatrix} \Theta_{11}(j;r) & \Theta_{12}(j;r) \\ \Theta_{21}(j;r) & \Theta_{22}(j;r) \end{Bmatrix} , \tag{84}$$

it follows from Eq. 83 that

$$\Theta_{11}(j) = \Theta_{11}(j-1) \tag{85a}$$

$$\Theta_{12}(j) = \Theta_{12}(j-1) \tag{85b}$$

$$\Theta_{21}(j) = -K_1(j) \cdot \Theta_{11}(j-1) + [1 - j \cdot K_1(j)] \cdot \Theta_{21}(j-1) \tag{86a}$$

$$\Theta_{22}(j) = -K_1(j) \cdot \Theta_{12}(j-1) + [1 - j \cdot K_1(j)] \cdot \Theta_{22}(j-1) \ , \tag{86b}$$

where for conciseness we temporarily suppress the second index (r) in the various Θ_{nm}. We find from Eqs. 27 and 85 that

$$\Theta_{11}(j;r) = 1 \ , \tag{87a}$$

$$\Theta_{12}(j;r) = 0 \ , \tag{87b}$$

and then from Eqs. 86 and 87 that

$$\Theta_{21}(j;r) = -K_1(j) + [1 - j \cdot K_1(j)] \cdot \Theta_{21}(j-1;r) \ , \tag{88a}$$

$$\Theta_{22}(j;r) = [1 - j \cdot K_v(j)] \cdot \Theta_{22}(j-1;r) \ . \tag{88b}$$

The solutions of Eqs. 88, with initial conditions given by Eq. 27, are developed in Appendix C as

$$\Theta_{21}(j;r) = \frac{3[r(r+1) - j(j+1)]}{j(j+1)(2j+1)} \ , \tag{89a}$$

$$\Theta_{22}(j;r) = \frac{r(r+1)(2r+1)}{j(j+1)(2j+1)} \ . \tag{89b}$$

Equations 87 and 89 provide the desired closed forms for the elements of Θ.

5.3 JUMP SIGNATURES

From Eqs. 24, 81, 82, and 89, we obtain

$$\begin{aligned} \begin{Bmatrix} F_{11}(j;r) & F_{12}(j;r) \\ F_{21}(j;r) & F_{22}(j;r) \end{Bmatrix} &= \sum_{n=r}^{J} K_v(n) \cdot \begin{Bmatrix} 1 & 0 \\ \Theta_{21} & \Theta_{22} \end{Bmatrix} \cdot \begin{Bmatrix} 0 & 0 \\ 1 & n \end{Bmatrix} \\ &= \sum_{n=r}^{J} K_v(n) \cdot \Theta_{22}(n;r) \cdot \begin{Bmatrix} 0 & 0 \\ 1 & n \end{Bmatrix} \ . \end{aligned} \tag{90}$$

From Eqs. 62 and 89b,

$$K_v(n) \cdot \Theta_{22}(n;r) = \frac{6n}{j(j+1)(2j+1)} \ . \tag{91}$$

From Eqs. 90 and 91,

$$F_{11}(j;r) = F_{12}(j;r) = 0 \tag{92}$$

and

$$F_{2m}(j;r) = \frac{6}{j(j+1)(2j+1)} \cdot \sum_{n=r}^{J} n^m \ , \quad m = 1,2 \ . \tag{93}$$

However, since

$$\sum_{n=1}^{J} n = \frac{1}{2} \cdot j(j+1) \ , \tag{94a}$$

$$\sum_{n=1}^{J} n^2 = \frac{1}{6} \cdot j(j+1)(2j+1) \ , \tag{94b}$$

and

$$\sum_{n=r}^{J} f(n) = \sum_{n=1}^{J} f(n) - \sum_{n=1}^{r-1} f(n) \ , \tag{94c}$$

it follows from Eqs. 92-94 that

$$F_{11}(j;r) = 0 , \tag{95a}$$

$$F_{12}(j;r) = 0 , \tag{95b}$$

$$F_{21}(j;r) = \frac{3}{j(j+1)(2j+1)} \cdot [j(j+1) - r(r-1)] , \tag{95c}$$

$$F_{22}(j;r) = \frac{1}{j(j+1)(2j+1)} \cdot [j(j+1)(2j+1) - r(r-1)(2r-1)] . \tag{95d}$$

From Eqs. 23, 40, and 81, it can be shown that

$$G_1(j;r) = 1 - j \cdot F_{21}(j-1;r) , \tag{96a}$$

$$G_2(j;r) = j \cdot [1 - F_{22}(j-1;r)] . \tag{96b}$$

It follows from Eqs. 95 and 96 that

$$G_1(j;r) = \frac{3r(r-1) - (j-1)(j+1)}{(j-1)(2j-1)} , \tag{97a}$$

$$G_2(j;r) = \frac{r(r-1)(2r-1)}{(j-1)(2j-1)} . \tag{97b}$$

Equations 95 and 97 are the desired closed forms for the elements of F and G, applicable when $j \geq r$; F and G are identically zero when $j < r$ (cf. Eq. 22).

5.4 STATE ESTIMATE AND INNOVATION JUMP RESPONSES

As noted in connection with Eqs. 20 and 21, the state estimate and innovation developed by the Kalman filter can be decomposed as

$$\hat{x}(k) = \hat{x}_1(k) + \hat{x}_2(j;r) , \tag{98a}$$

$$\gamma(k) = \gamma_1(k) + \gamma_2(j;r) , \tag{98b}$$

where we have defined

$$\hat{x}_2(j;r) \equiv F(k;\theta) \cdot \nu , \tag{99a}$$

$$\gamma_2(j;r) \equiv G(k;\theta) \cdot \nu . \tag{99b}$$

We refer to $\hat{x}_2$ and γ_2 as the state estimate jump response and innovation jump response, respectively.

We obtain from Eqs. 99, 95, 97, 78, and 48, for the state estimate jump response,

$$\hat{s}_2(j;r) = \frac{\alpha(j-r)[(j+1)-r][(2j+1)+r]}{j(j+1)(2j+1)} , \quad j \geq r , \tag{100}$$

and for the innovation jump response,

$$\gamma_2(j;r) = \frac{\alpha \cdot r(j^2 - r^2)}{(j-1)(2j-1)} , \quad j \geq r . \tag{101}$$

The jump responses are related by the equation

$$\begin{aligned} \gamma_2(j;r) &= z_2(j) - \hat{z}_2(j;r) \\ &= \alpha \cdot (j-r) - j \cdot \hat{s}_2(j-1;r) . \end{aligned} \tag{102}$$

The asysmptotic behavior of γ_2 is derived from Eq. 101 as

$$\lim_{j \to \infty} \gamma_2(j;r) = \alpha \cdot r/2 , \tag{103}$$

$$\lim_{\substack{j \to \infty \\ (j-r)/j \to 0}} \gamma_2(j;r) = \alpha \cdot (j-r) . \tag{104}$$

To verify the correctness of Eq. 101, we operated on a two-segment polyline (Fig. 6a) with our recursive regression algorithm, Eqs. 60 and 61. As shown in Fig. 6b, the difference between our closed-form expression for γ_2, Eq. 101, and the recursively developed innovation, Eq. 61, is imperceptible on the scale of the figure.

Equation 101 indicates that, in general,

$$\text{sgn}(\gamma_2) = \text{sgn}(\alpha) , \tag{105}$$

the sign of γ_2 is the same as the sign of α. In Fig. 7 we provide an example negative α. The dashed lines on Figs. 6b and 7b are the asymptotes given by qs. 103 and 104.

Citing numerical experience, McAulay and Denlinger[1] have assumed

$$\gamma_2(j;r) \approx K \cdot (j-r)^2 , \tag{106}$$

with K an unspecified constant, independent of j and r. (We have cast McAulay and Denlinger's Eq. 19 into our notation to facilitate comparison with our Eq. 101.) We have not explored the conditions under which McAulay and Denlinger's approximation for γ_2 will provide similar results to our exact expression, Eq. 101.

5.5 INVERSE COVARIANCE OF INNOVATION ESTIMATE

In this section we obtain closed-form expressions for the elements of the inverse covariance of $\hat{v}(k)$, $C(k;\theta)$, originally defined in Eq. 30.

With the definitions

$$C(k_0 + j; k_0 + r) \equiv \begin{Bmatrix} C_1(j;r) & C_2(j;r) \\ C_2(j;r) & C_3(j;r) \end{Bmatrix} \tag{107}$$

and

$$G(k_0 + j; k_0 + r) \equiv \begin{Bmatrix} G_1(j;r) \\ G_2(j;r) \end{Bmatrix} , \tag{108}$$

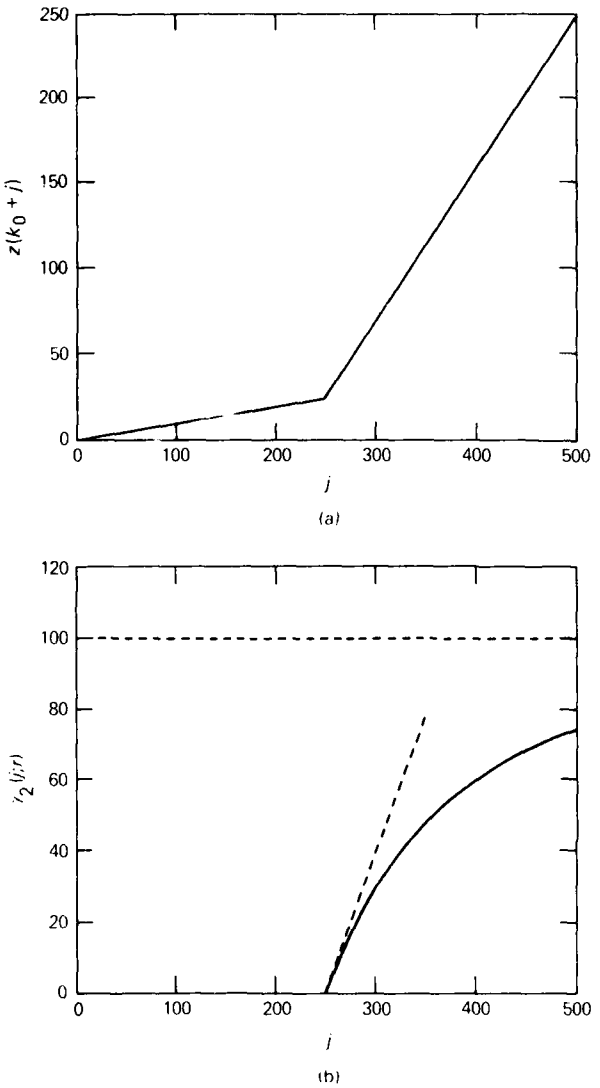


Figure 6 Numerical experiment validating the closed-form expression for innovation jump response, Eq. 101. (a) Two-segment polyline with parameters $(\alpha_0,\alpha,r) = (0.1,0.8,250)$ (cf. Fig. 3a). (b) Innovation jump response (solid line). The difference between the analytical expression, Eq. 101, and the innovation generated by recursive regression, Eq. 61, is imperceptible on the scale of the plot.

and with scalar $V(k)$ given by Eq. 63, it follows from Eq. 30 that

$$C_1(j;r) = \sum_{n=r}^{j} G_1^2(n;r)/V_0(n) \ , \tag{109a}$$

$$C_2(j;r) = \sum_{n=r}^{j} G_1(n;r) \cdot G_2(n;r)/V_0(n) \ , \tag{109b}$$

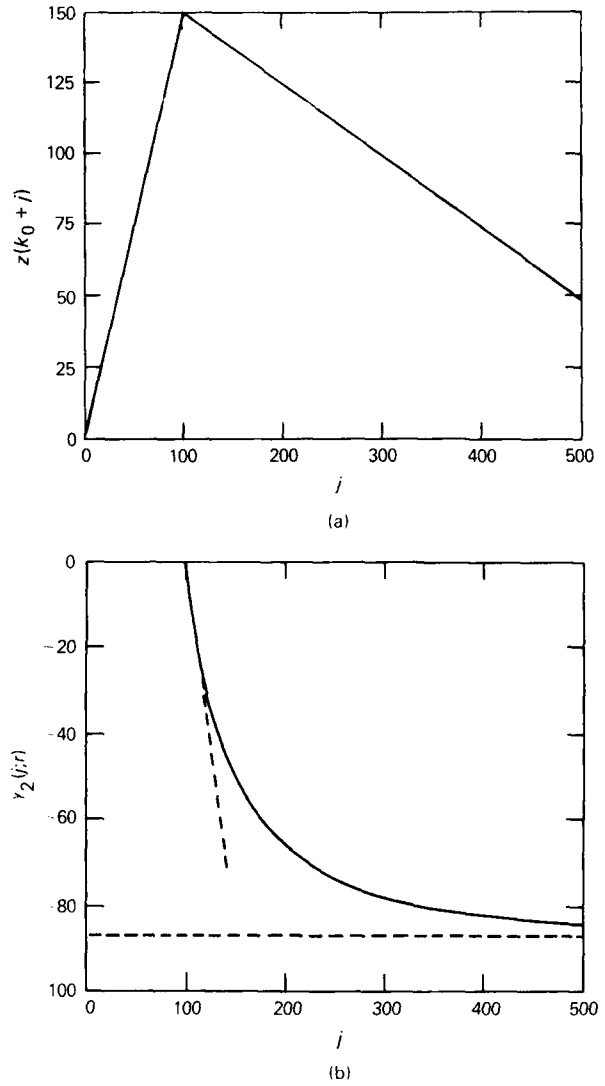


Figure 7 Numerical experiment validating the closed-form expression for innovation jump response, Eq. 101. (a) Two-segment polyline with parameters $(\alpha_0,\alpha,r) = (1.5, -1.75, 100)$ (cf. Fig. 3a). (b) Innovation jump reponse (solid line). The difference between the analytical expression, Eq. 101, and the innovation generated by recursive regression, Eq. 61, is imperceptible on the scale of the plot.

$$C_3(j;r) = \sum_{n=r}^{j} G_2^2(n;r)/V_0(n) \tag{109c}$$

Substituting Eqs. 97 for G_1 and G_2 and Eq. 63 for V_0, we find that Eqs. 109 can be put into the following form:

$$\sigma^2 \cdot C_1(j;r) = 3s_1 - [12r^2(r-1)^2 + 6r(r-1) + 1] \cdot s_2 + s_3 \quad , \tag{110a}$$

$$\sigma^2 \cdot C_2(j;r) = (2r-1) \cdot s_1 - r(r-1)(2r-1)^3 \cdot s_2 , \tag{110b}$$

$$\sigma^2 \cdot C_3(j;r) = \frac{1}{3} \cdot (2r-1)^2 \cdot [s_1 - 4r^2(r-1)^2 \cdot s_2] , \tag{110c}$$

where the quantities s_i are given by

$$s_1(j;r) \equiv r^2(r-1)^2 \cdot \sum_{n=r}^{j} 1/(n^2-1)$$

$$= \frac{1}{2} \cdot r(r-1)(2r-1) - \frac{r^2(r-1)^2(2j+1)}{2j(j+1)} , \tag{111a}$$

$$s_2(j;r) \equiv \sum_{n=r}^{j} 1/(4n^2-1)$$

$$= \frac{j}{2j-1} - \frac{r-1}{2r-1} , \tag{111b}$$

$$s_3(j;r) \equiv \sum_{n=r}^{j} n^2/(4n^2-1)$$

$$\approx \frac{1}{4} \cdot (j-r+1) + s_2(j;r) . \tag{111c}$$

Equations 110 and 111 provide the desired closed forms for the elements of $C(k;\theta)$. Although $C(k;\theta)$ can, in principle, be computed off line and stored, it is actually not necessary to do this, as we will show in Section 5.7.

5.6 THE FILTERED INNOVATION

In this section we provide simple recursive filters for developing the elements of $d(k;\theta)$, the filtered innovation, originally defined in Eq. 29.

With the definition

$$d(k_0 + j; k_0 + r) = \begin{Bmatrix} d_1(j;r) \\ d_2(j;r) \end{Bmatrix} , \tag{112}$$

we find from Eqs. 29, 63, and 108 that

$$\sigma^2 \cdot d_1(j;r) = \sum_{n=r}^{j} \gamma(k_0+n) \cdot [\sigma^2 \cdot G_1(n;r)/V_0(n)]$$

$$\sigma^2 \cdot d_2(j;r) = \sum_{n=r}^{j} \gamma(k_0+n) \cdot [\sigma^2 \cdot G_2(n;r)/V_0(n)] \tag{113}$$

Substituting Eqs. 97 for G_1 and G_2 and Eq. 63 for V_0 in Eq. 113, we obtain

$$\sigma^2 \cdot d_1(j;r) = \sum_{n=r}^{j} \gamma(k_0+n) \cdot A(n;r)$$

$$\sigma^2 \cdot d_2(j;r) = \sum_{n=r}^{j} \gamma(k_0+n) \cdot B(n;r) \tag{114}$$

where

$$A(n;r) \equiv \frac{3r(r-1) - (n^2-1)}{(n+1)(2n+1)}$$
$$B(n;r) \equiv \frac{r(r-1)(2r-1)}{(n+1)(2n+1)} \quad . \tag{115}$$

With the definitions

$$D_1(j;i) \equiv \sigma^2 \cdot d_1(j;j-i+1)$$
$$D_2(j;i) \equiv \sigma^2 \cdot d_2(j;j-i+1) \quad , \tag{116}$$

it follows from Eq. 114 that

$$D_1(j;i) = D_1(j-1;i-1) + A(j;j-i+1) \cdot \gamma(k_0+j)$$
$$D_2(j;i) = D_2(j-1;i-1) + B(j;j-i+1) \cdot \gamma(k_0+j) \quad . \tag{117}$$

Equations 117 are the desired recursion relations for generating the filtered innovation sequences d_1 and d_2. Equations 117 are used as follows. When a new measurement is received at time j, the one-parameter regression generates via Eq. 61 a new innovation value, $\gamma(k_0+j)$. Our interest is in obtaining the values of $d_1(j;r)$ and $d_2(j;r)$ for the M "most recent" values of r, where, to make our example concrete, we select $M = 20$ (cf. Fig. 2b). Thus, our interest is in obtaining $d_1(j;r)$ and $d_2(j;r)$ for $r = j, (j-1), \ldots, (j-19)$. With Eq. 116, we formulate our problem as

$$\begin{aligned} &\text{Given} \quad \{D_1(j-1;i) \ , \ i = 1,2,\ldots,20\} \text{ and } \gamma(k_0+j) \\ &\text{Calculate} \quad \{D_1(j;i) \ , \ i = 1,2,\ldots,20\} \end{aligned} \tag{118}$$

For $i = 1$, we obtain from Eq. 117

$$D_1(j;1) = D_1(j-1;0) + A(j;j) \cdot \gamma(k_0+j) \quad . \tag{119}$$

From Eq. 116,

$$D_1(j-1;0) = \sigma^2 \cdot d_1(j-1;j) \quad , \tag{120}$$

and from Eqs. 22 and 29

$$d_1(j;r) = 0 \ , \ j < r \ ,$$

so that

$$D_1(j-1;0) = 0 \quad . \tag{121}$$

The solution to our problem, Eq. 118, follows from Eqs. 117, 119, and 121:

$$\begin{aligned} D_1(j;1) &= A(j;j) \cdot \gamma(k_0 + j) \\ D_1(j;2) &= A(j;j-1) \cdot \gamma(k_0 + j) + D_1(j-1;1) \\ D_1(j;3) &= A(j;j-2) \cdot \gamma(k_0 + j) + D_1(j-1;2) \\ &\vdots \\ D_1(j;20) &= A(j;j-19) \cdot \gamma(k_0 + j) + D_1(j-1;19) \end{aligned} \tag{122}$$

The solution for D_2 is obtained by making the replacements

$$D_2 \leftarrow D_1$$

$$B \leftarrow A$$

in Eq. 122.

Although the coefficients $A(n;r)$ and $B(n;r)$ given by Eq. 115 could in principle be calculated off line, we chose in our implementation of the GLR algorithm to alleviate storage requirements (at some cost in execution speed) by calculating the coefficients as

$$\begin{aligned} A(n;r) &= R_a(r) \cdot N_b(n) - N_a(n) \\ B(n;r) &= R_b(r) \cdot N_b(n) \end{aligned} \tag{123}$$

The vectors R_a, N_a, R_b, and N_b were calculated off line and made available to the algorithm, where

$$R_a(r) \equiv 3r(r-1) , \tag{124a}$$

$$N_a(n) \equiv \frac{n-1}{2n+1} , \tag{124b}$$

$$R_b(r) \equiv r(r-1)(2r-1) , \tag{124c}$$

$$N_b(n) \equiv \frac{1}{(n+1)(2n+1)} . \tag{124d}$$

In our GLR implementation we calculated 200 values of each of the four vectors defined in Eq. 124, for a total of 800 stored coefficients. Calculating the complete set of coefficients $A(n;r)$ and $B(n;r)$ off line, rather than computing them as needed via Eq. 123, would have required storing $2 \times 200 \times 200 = 80{,}000$ coefficients.

5.7 GENERALIZED LIKELIHOOD RATIO

From Eqs. 37b, 50, and 112,

$$b(k_0 + j;k_0 + r) = d_2(j;r) - r \cdot d_1(j;r) . \tag{125}$$

From Eqs. 37a, 50, and 107,

$$a(k_0 + j;k_0 + r) = r^2 \cdot C_1(j;r) - 2r \cdot C_2(j;r) + C_3(j;r) . \quad (126)$$

The likelihood ratio is then obtained from Eq. 36 as

$$\ell(k;\theta) = b^2(k;\theta)/a(k;\theta) . \quad (36)$$

At every time increment (i.e., when k increases by unity), Eq. 36 is used to compute $M = 20$ new values of ℓ:

$$\ell(k;k - i + 1) , \quad i = 1,2, \ldots, 20 .$$

Recursive relations, Eqs. 117, are used to establish d_i, which in turn enter into the numerator of Eq. 36 by means of Eq. 125. The C_i coefficients, Eqs. 110, are used in off-line calculations of $a(k;\theta)$ via Eq. 126, which in turn enter into the denominator of Eq. 36.

The GLR algorithm flow of operations is shown in Fig. 2b.

6.0 NUMERICAL EXPERIENCE

6.1 INTRODUCTION

In this section we present some numerical results illustrating the effectiveness of several polyline algorithms. The algorithms evaluated numerically are listed in Table 2, grouped according to speed of execution.

Performance in all cases was based on approximation of the infrared cloud/sky waveform shown in Fig. 8a.[22,23] The various regions of the waveform (broken clouds, blue sky, etc.) were identified by eye and labeled manually. However, one objective of the current work is to facilitate the development of algorithms capable of automatically segmenting this type of waveform data. An illustrative polyline representation of the data is shown in Fig. 8b, as provided by algorithm GLR-M (discussed in Section 6.5).

Comparison between generally similar waveforms (e.g., Figs. 8a and 8b) is facilitated by the use of numerical measures of similarity, or "error metrics." Our selection of error metrics (for approximations with a fixed number of knots, i.e., for fixed data compaction) includes:

[22]R. A. Steinberg and M. J. McHugh, *An Error Detection and Smoothing Algorithm for Infrared Data*, JHU/APL TG 1355 (Apr 1986).
[23]L. M. Howser, *Wide Area Guidance and Control Program: Investigation of Scanning IR Seeker Performance in Background Clutter*, JHU/APL TG 1360 (Dec 1986).

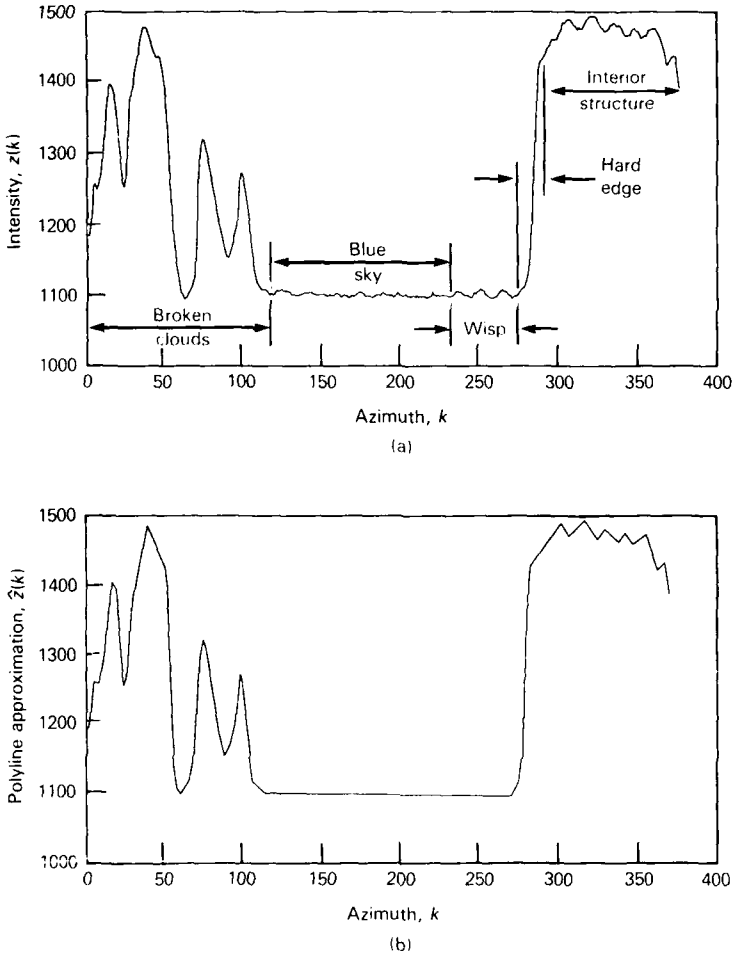


Figure 8 (a) Infrared cloud/sky waveform used in testing polyline algorithms. (b) Polyline approximation composed of 49 segments, generated by algorithm GLR-M.

- Average error, E_0
- Average absolute error, E_1
- Root mean square error, E_2
- Maximum absolute error, E_∞
- Execution speed, T.

All calculations were performed on an IBM PC AT computer.

Table 2
Polyline algorithms evaluated numerically.

	Algorithm	Author/Reference	Method
Slow			
	EMK	McLaughlin and Zacharski[9]	E-maximal knots
	DP	Dunham[11] and Papakonstantinou[10]	Dynamic programming
	GLR	Steinberg (Sect. 5)	Generalized likelihood ratio
Fast			
	LSBPT	Tomek[2]	Longest segment between parallel tangents
	CIM	Williams[4] and Sklansky and Gonzalez[5]	Cone intersection method
	HOP-F	Pavlidis[7]	Hop-along algorithm (fast)
	HOP-S	Pavlidis[7]	Hop-along algorithm (slow)
	BAD	Wall and Daniellson[16]	Bounded area deviation
	RLS1	Steinberg (Sect. 4, Eq. 73)	Recursive least squares (fast)
	RLS2	Steinberg (Sect. 4, Eq. 74)	Recursive least squares (slow)

6.2 A NOTE ON ERROR METRICS AND PERFORMANCE COMPARISONS

When the infrared camera used to obtain Fig. 8a is operated under closed-cover conditions (with the lens cap on), the output waveforms are reduced to zero-mean, unit-variance Gaussian noise. In fact, all data obtained with this camera, including Fig. 8a, have an additive Gaussian noise component of one count root mean square. In this section we discuss how additive Gaussian noise of known level affects the relative usefulness of alternative error metrics.

It is common in the literature to select peak error, E_∞, as the primary measure of polyline algorithm performance. Defining K as the number of knots in the approximation and P as the number of points in the unapproximated digital waveform, we obtain

$$\lim_{K \to P} \quad E_\infty = 0 \ , \tag{126}$$

assuming that knot locations are required to coincide with the waveform's sample values, a requirement commonly imposed by designers of algorithms that minimize E_∞. At fixed data compaction (i.e., for a fixed value of K), we can use E_∞ to compare the goodness of fit provided by alternative algorithms; the algorithms are ranked according to how close E_∞ comes to the zero ideal value.

An alternative approach, which we adopt in the present work, is to select E_2 as the primary performance metric. Ideal performance is expressed as

$$E_2 = 1 \ , \tag{127}$$

because the data are known to contain an additive component of zero-mean, unit-variance Gaussian noise. If, corresponding to approximations A and B, we have root mean square fitting errors $E_{2,A}$ and $E_{2,B}$, and if

$$0 < E_{2,A} < E_{2,B} < 1 \ , \tag{128}$$

we recognize the B approximation as being better than the A approximation; both approximations are under-smoothing the data and, very likely, using more knots than are really needed to represent the underlying structure.

6.3 AN EXPERIMENT AT 7:1 DATA COMPACTION

The ten algorithms listed in Table 2 were used to develop 50-knot approximations to Fig. 8a, corresponding roughly to 7:1 data compaction. The results are summarized in Table 3.

Table 3
Performance of polyline algorithms in developing 50-knot approximations to an infrared cloud/sky waveform, Fig. 8a (data compaction ≈ 7:1).

		Error			
Algorithm	Execution Time (s)	E_0 Average	E_1 Avg. Abs.	E_2 rms	E_∞ Peak
EMK	34.02	−0.81	2.95	3.74	8.80
DP	12.14	−0.36	2.95	3.69	**8.10**
GLR	1.79	−0.05	**2.24**	**3.00**	11.24
LSBPT	0.24	−1.68	4.26	4.87	11.75
CIM	0.35	−0.81	2.95	3.74	8.80
HOP-F	0.24	0.23	2.61	3.59	11.62
HOP-S	0.54	0.13	2.60	3.57	10.70
BAD	0.17	−1.82	4.21	4.95	13.77
RLS1	**0.14**	**−0.01**	2.76	3.40	9.26
RLS2	0.21	−0.01	2.76	3.40	9.26

Inspection of Table 3 shows that DP had the smallest peak error, while GLR had the smallest root mean square and average absolute errors. RLS and GLR had, by far, the smallest average error. The two fastest methods were RLS1 and BAD, with RLS enjoying, relative to BAD, a small advantage in speed and large advantages in every measure of fitting accuracy.

Although McLaughlin and Zacharski claim that EMK "...allows...one to optimally approximate the data...",[9] it appears from Table 3 that this is not so. Although EMK yields excellent results, its approximation is generally very similar to that provided by CIM, while CIM executes about 100 times faster.

The slow and fast variants of RLS were found in this case to yield the same approximation, although in our later tests this was found not to be generally true.

Tomek's algorithm, LSBPT, is of academic and historical interest as perhaps the first linear-time polyline algorithm derived from geometrical reasoning (as contrasted with, for example, the statistical and mathematical frameworks of least-squares and spline approximations). While LSBPT is very similar in structure to the subsequently developed CIM, Table 3 indicates that LSBPT executes about 50% faster than CIM but provides significantly poorer fitting accuracy. RLS is both faster and more accurate than LSBPT.

I note that Tomek[2] describes both a fast and a slow variant of his algorithm. Both of Tomek's algorithms were implemented, and it was found that the slow al-

gorithm offered almost no improvement in fitting accuracy. I consequently report only the results obtained with Tomek's fast algorithm.

The fast variant of Pavlidis's hop-along algorithm, HOP-F, offers significantly faster speed than HOP-S, for a relatively small penalty in fitting accuracy.

Our preliminary assessment based on Table 3 is as follows:

1. DP, while very slow, provides smallest peak error, E_∞.
2. GLR, provides the smallest root mean square error, E_2, and average absolute error, E_1.
3. RLS provides very large improvements in speed relative to GLR and DP, at relatively low cost in degraded fitting accuracy.

6.4 A CLOSER LOOK AT THE GLR APPROXIMATION

The statistics given in Table 3 are useful as an overall performance summary. In this section we present additional ways of examining performance intended to provide a more detailed and intuitively meaningful picture of algorithm strengths and weaknesses. We confine the discussion in this section to GLR; similar details on the performance of several other algorithms are provided in Appendix D.

In Fig. 9a we show the original waveform with the polyline knots superimposed (each knot is denoted with a + symbol). While the polyline approximation actually comprises a set of line segments connecting the knots, we have elected not to show the line segments in order to avoid excess detail in the figure. In Fig. 9b we show the pointwise difference between the original waveform and the polyline approximation, i.e., the pointwise fitting error. The histogram of the pointwise error (Fig. 9c) is nearly Gaussian, with several outliers in one of the tails of the density.

Since it is common in the literature to see emphasis on the E_∞ metric, we presently consider the performance of GLR where its peak error is worst, viz., in the neighborhoods identified in Fig. 9 by the circled numbers 1-3. These peak-error neighborhoods are expanded (zoomed) in Fig. 10; Table 4 presents the data from Fig. 10 in tabular form.

Table 4
Data in the neighborhoods of the three points of worst fit for the GLR algorithm (data are plotted in Fig. 10).

k	$z(k)$	Knot	$\hat{z}(k)$	$(z - \hat{z})$	Max. Error
22	1336		1335.91	0.09	
23	1296		1298.60	− 2.60	
24	1263	8	1261.30	1.70	
25	1254	9	1254.00	0.00	
26	**1267**		**1278.24**	**−11.24**	**←1**
27	1293		1302.47	− 9.47	
28	1325		1326.71	− 1.71	
29	1355		1350.95	4.05	
30	1379	10	1375.18	3.82	
275	1106		1108.48	− 2.48	
276	1115	35	1111.64	3.36	
277	1128		1131.05	− 3.05	
278	1152	36	1150.47	1.53	
279	**1200**		**1211.18**	**−11.18**	**←2**
280	1269		1271.89	− 2.89	

Table 4 (Continued)
Data in the neighborhoods of the three points of worst fit for the GLR algorithm (data are plotted in Fig. 10).

k	$z(k)$	Knot	$\hat{z}(k)$	$(z - \hat{z})$	Max. Error
281	1340		1332.60	7.40	
282	1392	37	1393.31	- 1.31	
283	1416		1411.72	4.28	
8	1252		1256.10	- 4.10	
9	1257	4	1255.30	1.70	
10	1270		1272.92	2.92	
11	1283		1290.54	7.54	
12	**1298**		**1308.17**	**- 10.17**	**-3**
13	1319		1325.79	- 6.79	
14	1340		1343.42	- 3.42	
15	1364		1361.04	2.96	
16	1384		1378.67	5.33	

6.5 A HEURISTIC MODIFICATION TO GLR

Perhaps the most important observation that can be made concerning Fig. 9a is that GLR unnecessarily uses a large number of knots to represent the unstructured blue sky portion of the waveform ($120 < k < 230$). By representing this benign portion of the data more efficiently, we can free up additional knots needed to represent the data more accurately in the highly structured parts of the waveform. This suggested that we develop a patch to our GLR routine (i.e., additional code to detect the blue sky region) and implement the RLS2 algorithm rather than GLR in this region.

We observe from Fig. 9a that the three largest errors are located on steep shoulders, i.e., regions of the waveform where the slope is relatively large. This suggests the following second ad hoc modification to our GLR algorithm: when the slope is larger than some threshold value, ϵ_s, replace the nominal likelihood threshold ϵ (Fig. 2) by a smaller number $\kappa \cdot \epsilon$, where $0 < \kappa < 1$, i.e.,

$$\epsilon \rightarrow \begin{cases} \epsilon & , \quad \hat{s}(j) < \epsilon_s \\ \kappa \cdot \epsilon & , \quad \hat{s}(j) \geq \epsilon_s \end{cases} \tag{129}$$

where $\hat{s}(j)$ is the current estimate of line slope, and $j \equiv (k - k_0)$ is the distance from the previous knot. An appropriate value for ϵ_s may be developed from the data in Table 4.

We refer to the resulting modified version of GLR, which incorporates both ad hoc patches, as GLR-M. The 50-knot approximation obtained with GLR-M is shown in Fig. 11, analogous to our earlier Fig. 9 results for GLR. Comparing Fig. 11a with Fig. 9a we note that the blue sky region is represented more efficiently by GLR-M; comparing Fig. 11c with Fig. 9c we see that the fitting error histogram is more compact for GLR-M than for GLR.

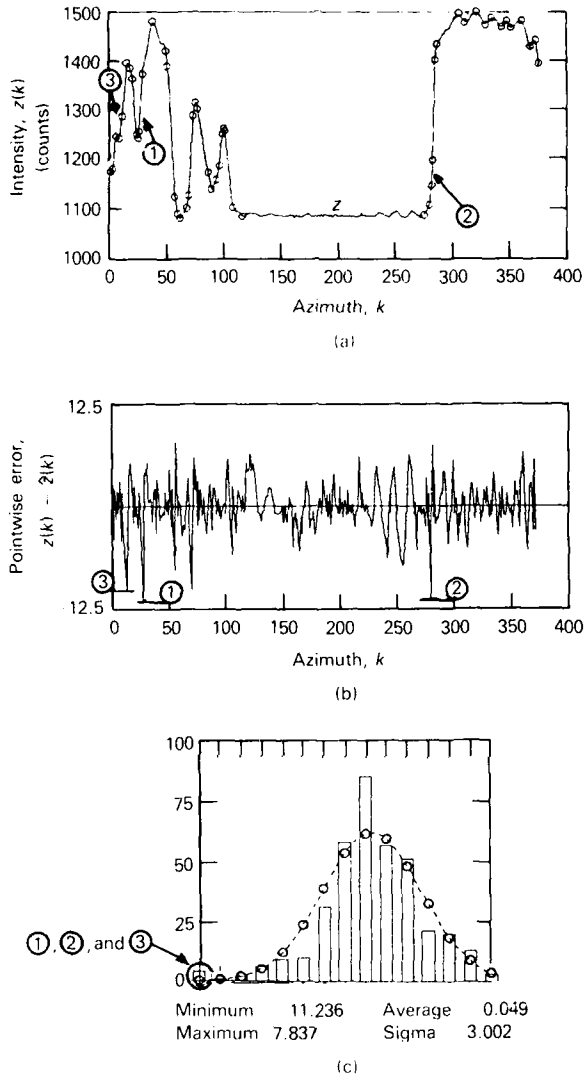


Figure 9 GLR polyline algorithm evaluation. (a) IR cloud waveform (solid curve, z) and polyline knots (+). (b) Pointwise difference between z and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

In Table 5 we compare GLR-M to DP and to our original GLR routine. Compared with the other algorithms, GLR-M displays significantly superior performance with respect to E_1 and E_2, while simultaneously achieving a value of E_∞ just 4% worse than the dynamic programming solution.

The patch applied to GLR was tuned specifically to achieve a 50-knot approximation; i.e., GLR-M is not sufficiently robust to be directly comparable to the other algorithms (which is why GLR-M results were not included in Table 3). Nonetheless, our experiments with GLR-M suggest that, with additional effort, it may be possible to achieve excellent fitting simultaneously in all error metrics.

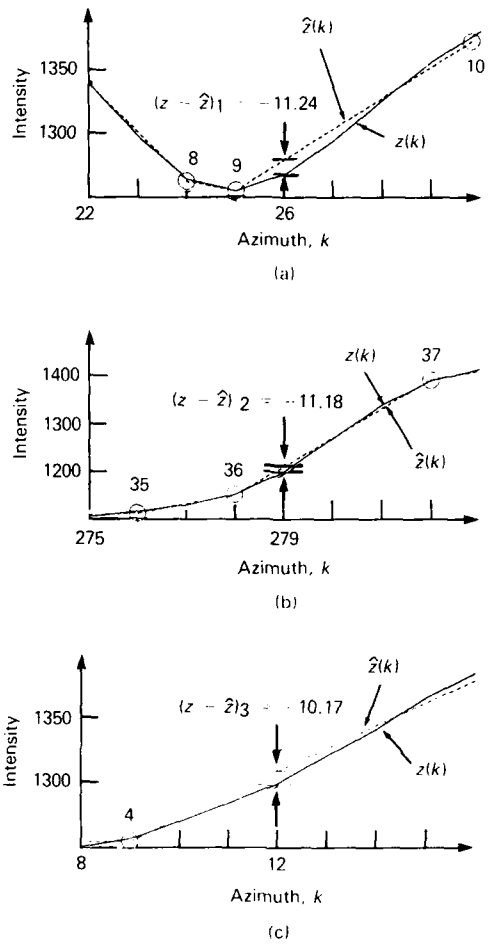


Figure 10 Data in the neighborhoods of the three points of worst fit for the GLR algorithm. Figure 9a shows these neighborhoods within their larger context; Table 4 provides these data in tabular form. (a) Neighborhood of peak error point. (b) Neighborhood of second worst point. (c) Neighborhood of third worst point.

Table 5
Performance of three polyline algorithms. Results for DP and GLR as for Table 3; GLR-M is a heuristically modified version of GLR.

		Error			
Algorithm	Execution Time (s)	E_0 Average	E_1 Avg. Abs.	E_2 rms	E_∞ Peak
DP	12.14	−0.36	2.95	3.69	8.10
GLR	1.79	−0.05	2.24	3.00	11.24
GLR-M	1.70	0.13	2.17	2.77	8.45

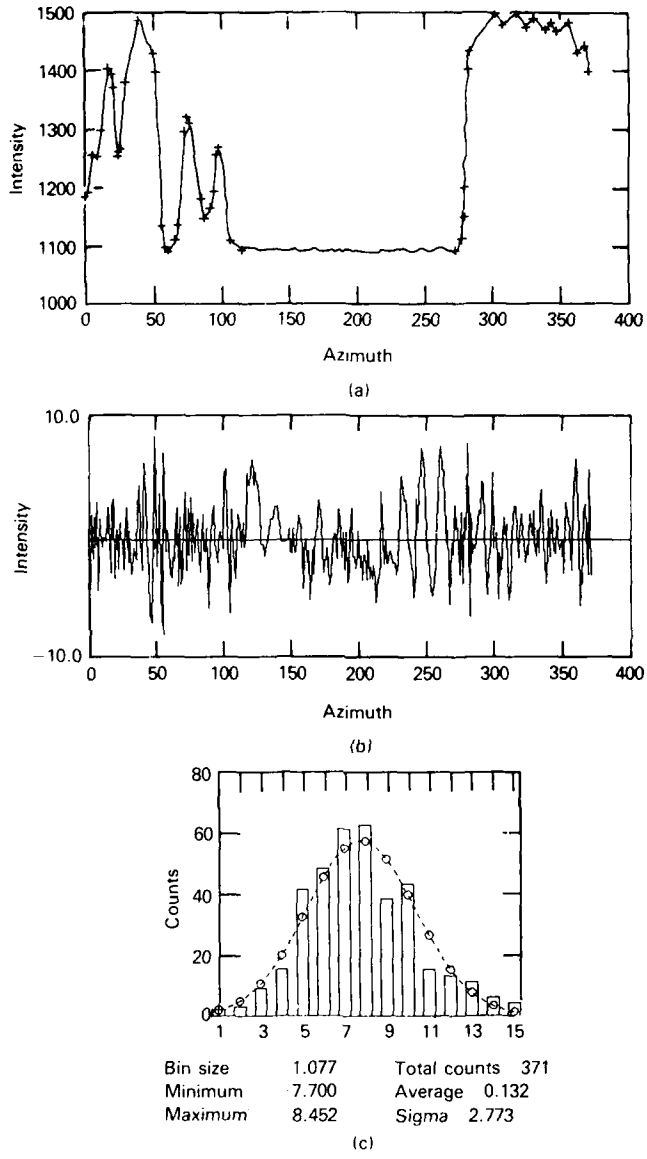


Figure 11 GLR-M polyline algorithm evaluation. (a) IR cloud waveform (solid curve) and polyline knots (o). (b) Pointwise fitting error. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

6.6 ROBUSTNESS WITH RESPECT TO DATA COMPACTION

All results reported in Sections 6.3 through 6.5 have been for 50-knot approximations to Fig. 8a, corresponding to a fixed $\approx 7{:}1$ data compaction. In this section we present results of algorithm performance over a range of data compactions, viz., approximations of 40–70 knots. Our approach is to display the variation of each performance metric (E_0, E_1, E_2, E_∞, and T) with the number of knots, for each algorithm.

Of the various algorithms described in the literature, CIM provides perhaps the best combination of good fitting accuracy, fast execution speed, robustness, and ease of use. For these reasons, the performances of the other algorithms are in every case compared with that of the CIM baseline.

Results presented in this section compare GLR with CIM (Fig. 12) and RLS1 with CIM (Fig. 13). Analogous figures for the other algorithms are provided in Appendix E. In all cases, the dashed curves are for CIM.

The plots in Figs. 12f and 13f refer to the threshold parameter ϵ as it appears in Eq. 73 for RLS1 and in Fig. 2b for GLR. The parameter ϵ for CIM is simply the peak error:

$$\epsilon = E_\infty \ , \quad \text{for GLR} \ . \tag{130}$$

Thus, as previously noted by Dunham,[1] CIM has an advantage in ease of use relative to algorithms (such as RLS and GLR) with ϵ parameters that do not directly control an error metric.

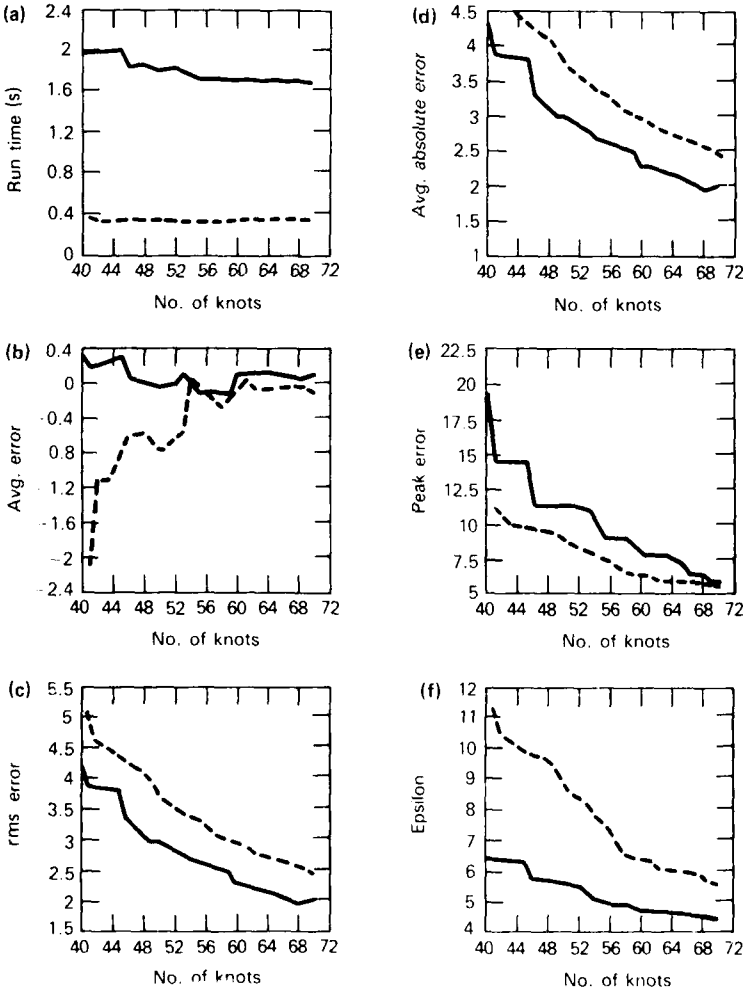


Figure 12 Performance statistics as functions of data compaction for GLR (solid curves) and CIM (dashed curves).

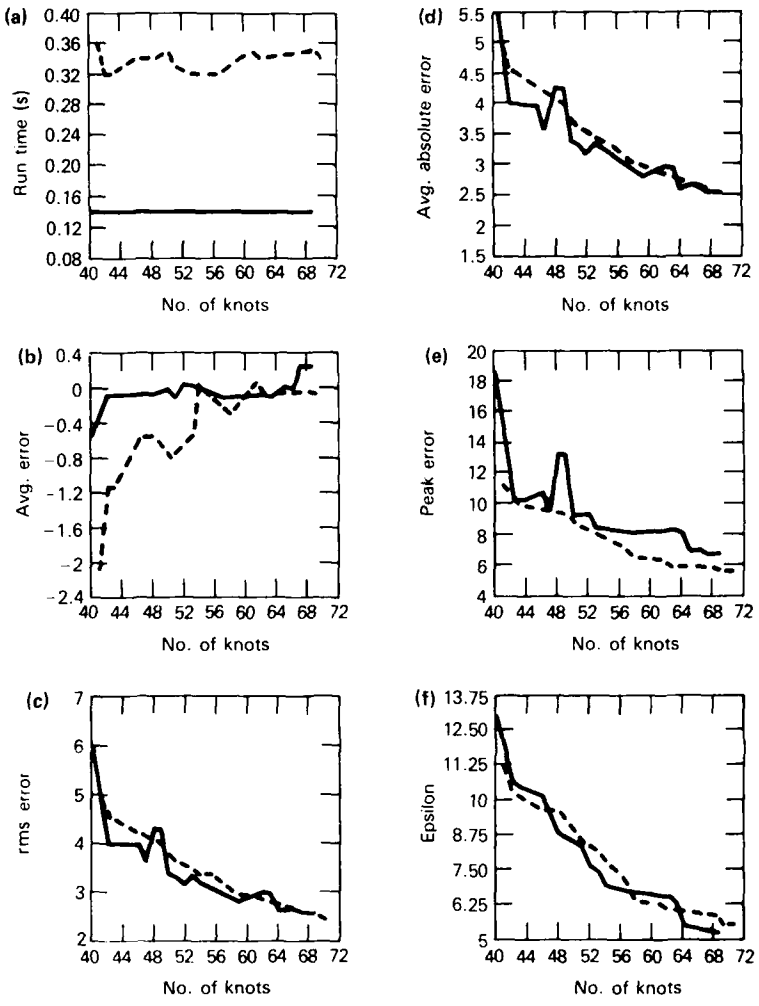


Figure 13 Performance statistics as functions of data compaction for RLS1 (solid curves) and CIM (dashed curves).

7.0 CONCLUSIONS

The extraction of shape information from waveforms can be facilitated by preprocessing the waveform with a polyline algorithm, i.e., an algorithm that approximates the waveform as a concatenated sequence of straight-line segments. Polyline approximation accomplishes smoothing of small-amplitude structure while leaving large-amplitude structure well defined and unsmoothed. Moreover, the polyline representation generally provides a large degree of data compaction relative to the original time series.

The new polyline approximation algorithms developed in this report are of two types, which we refer to as recursive least squares (RLS) and generalized likelihood ratio (GLR) algorithms. The performance of these algorithms is assessed in Section 6 by comparison with a number of alternative approaches. The Cone Intersection Method (CIM) algorithm discovered independently by Williams[4] and by Sklansky and Gonzalez[5] is used as a benchmark against which to compare the performance of other methods, including our own RLS and GLR algorithms.

The relative performance of the various algorithms turns out to be somewhat dependent on the choice of error metric. In particular, the original algorithms presented here are considerably superior to prior algorithms, with respect to bias and root mean square error.

Our RLS algorithms derived in Section 4 are based on a simple Kalman filter. Our numerical experience is that, relative to the best of the prior approaches, RLS has (Figs. 13 and E-6)

- Faster execution speed
- Slightly worse peak error
- Slightly superior root mean square error

RLS enjoys a significant speed advantage relative to the other algorithms we have evaluated.

Our GLR polyline algorithm is derived in Section 5 as an application of a formalism originally provided by Willsky and Jones.[17] Some characteristics of GLR, relative to the best of the prior approaches, are (Fig. 12)

- Worse peak absolute error
- Superior root mean square error

Although most of our results indicate a trade-off between root mean square fitting error (E_2) and peak error (E_∞), our numerical experiments with a currently nonrobust GLR variant indicate that it may be possible to obtain excellent fitting in all metrics simultaneously.

Conclusions regarding other approaches are as follows:

1. The method of E-maximal knots[9] yields nearly identical fitting performance to that of CIM but is about 100 times slower in execution (Fig. E-1 in Appendix E).
2. The dynamic programming method of Refs. 10 and 11 provides a small advantage in E_∞ and nearly equal E_2 relative to CIM but is about 30 times slower in execution (Fig. E-2).
3. Tomek's algorithm[2] executes about 50% faster than CIM but provides substantially worse fitting accuracy both in E_2 and in E_∞ (Fig. E-3).
4. Pavlidis's hop-along algorithm[7] executes about 50% slower than CIM, is slightly superior in E_2, and significantly worse in E_∞ (Fig. E-4).
5. The bounded-area-deviation algorithm[16] is only slightly slower than RLS1; however, it provides significantly worse values of E_2 and E_∞ than RLS1 (Figs. E-5 and E-7).

ACKNOWLEDGMENTS

Thanks are due to W. J. Tropf, APL/F1F, and the members of the APL Independent Research and Development (IRAD) Committee for supporting this work. The infrared cloud/sky data were obtained from the Navy's Background Measurements and Analysis Program, Naval Research Laboratory, Washington, D.C.

REFERENCES

[1] R. J. McAulay and E. J. Denlinger, "A Decision-Directed Adaptive Tracker," *IEEE Trans. Aerosp. Electron. Syst.* **AES-9**, 229–236 (1973).

[2] I. Tomek, "Two Algorithms for Piecewise Linear Continuous Approximation of Functions of One Variable," *IEEE Trans. Comput.* **C-23**, 445–448 (1974).

[3] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin (1977).

[4] C. M. Williams, "An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves," *Comput. Graph. Image Process.* **8**, 286–293 (1978).

[5] J. Sklansky and V. Gonzalez, "Fast Polygonal Approximation of Digitized Curves," in *Proc. 1979 IEEE Computer Society Conf. Pattern Recognition Image Processing*, pp. 604–609 (1979).

[6] T. Pavlidis and S. L. Horowitz, "Segmentation of Plane Curves," *IEEE Trans. Comput.* **C-23**, 860–870 (1974).

[7] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Md. (1982).

[8] J. Vandewalle, "On the Calculation of the Piecewise Linear Approximation to a Discrete Function," *IEEE Trans. Comput.* **C-24**, 843–846 (1975).

[9] H. W. McLaughlin and J. J. Zacharski, "Segmented Approximation," in *Approximation Theory*, E. W. Cheney, ed., Academic Press, New York, pp. 647–654 (1980).

[10] G. Papakonstantinou, "Optimal Polygonal Approximation of Digital Curves," *Signal Process.* **8**, 131–135 (1985).

[11] J. G. Dunham, "Optimum Uniform Piecewise Linear Approximation of Planar Curves," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 67–75 (1986).

[12] F. Badi'i and B. Peikari, "Functional Approximation of Planar Curves via Adaptive Segmentation," *Int. J. Systems Sci.* **13**, 667–674 (1982).

[13] U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Comput. Graph. Image Process.* **1**, 244–256 (1972).

[14] J. Roberge, "A Data Reduction Algorithm for Planar Curves," *Comput. Vision Graph. Image Process.* **29**, 168–195 (1985).

[15] T. Pavlidis, "Editorial—Papers on Shape Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1 (1986).

[16] K. Wall and P. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves," *Comput. Graph. Image Process.* **28**, 220–227 (1984).

[17] A. S. Willsky and H. L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," *IEEE Trans. Auto. Control* **AC-21**, 108–112 (1976).

[18] S. F. Haase, "Advanced Radar Tracking Techniques," in *IR&D/B&P Program Plan Vol. II FY1988*, JHU/APL, Laurel, Md., pp. II-375 to II-397 (1988).

[19] A. Madiwale and B. Friedland, "Comaprison of Innovations-Based Analytical Redundancy Methods," in *Proc. 1983 Am. Control Conf.*, pp. 940–945 (1983).

[20] M. Basseville and A. Benveniste, "Design and Comparative Study of Some Sequential Jump Detection Algorithms for Digital Signals," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-31**, 521–535 (1983).

[21] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press (1988).

[22] R. A. Steinberg and M. J. McHugh, *An Error Detection and Smoothing Algorithm for Infrared Data*, JHU/APL TG 1355 (Apr 1986).

[23] L. M. Howser, *Wide Area Guidance and Control Program: Investigation of Scanning IR Seeker Performance in Background Clutter*, JHU/APL TG 1360 (Dec 1986).

BIBLIOGRAPHY

Application	References
Approximation of mathematical functions	24–28
Encoding of medical information	
Electrocardiagram waveforms	3, 6, 8, 10, 29–31
Rib and lung boundaries in radiographs	5, 11, 32
Cell and chromosome boundaries	3, 6, 11, 13, 33, 34
Statistical data analysis	
Market analysis of new housing starts	35
Histories of U.S. flour prices and number of U.S. telephones	36
Weight/height ratio of preschool children	37
Boundary encoding for computer vision and graphics	
General	2, 4, 7, 14, 38–42
Coastlines	11, 13, 16, 34, 43
Alphanumeric symbols	16, 34, 44, 45
Terrain profiles	29
Hand tools	43, 46
Aircraft silhouettes	12
Miscellaneous	
Aircraft tracking	1, 18
Electron microscope profiles	29
Stagnant surface layer height in flowing water	47

[24] J. R. Rice, "On Adaptive Piecewise Polynomial Approximation," in *Theory of Approximation*, A. G. Law and B. N. Sahney, eds., Academic Press, New York, pp. 359–386 (1976).

[25] J. Chow, "On the Uniqueness of Best $L_2[0,1]$ Approximation by Piecewise Polynomials with Variable Breakpoints," *Math. Comput.* **39**, 571–585 (1982).

[26] G. M. Phillips, "Algorithms for Piecewise Straight Line Approximations," *Comput. J.* **11**, 211–212 (1968).

[27] A. Cantoni, "Optimal Curve Fitting with Piecewise Linear Functions," *IEEE Trans. Comput.* **C-20**, 59–67 (1971).

[28] H. Stone, "Approximation of Curves by Line Segments," *Math. Comput.* **15**, 40–47 (1961).

[29] T. Pavlidis, "Waveform Segmentation Through Functional Approximation," *IEEE Trans. Comput.* **C-22**, 689–697 (1973).

[30] F. Gritzali and G. Papakonstantinou, "A Fast Piecewise Linear Approximation Algorithm," *Signal Process.* **5**, 221–227 (1983).

[31] S. L. Horowitz, "Peak Recognition in Waveforms," in *Syntactic Pattern Recognition Applications*, K. S. Fu, ed., Springer-Verlag, Berlin, pp. 31–49 (1977).

[32] J. Sklansky and V. Gonzalez, "Fast Polygonal Approximation of Digitized Curves," *Pattern Recognition* **12**, 327–331 (1980).

[33] T. Pavlidis, "Polygonal Approximations by Newton's Method," *IEEE Trans. Comput.* **C-26**, 800–807 (1977).
[34] J. C. Bezdek and I. M. Anderson, "An Application of the c-Varieties Clustering Algorithms to Polygonal Curve Fitting," *IEEE Trans. Syst. Man Cybern.* **SMC-15**, 637–641 (1985).
[35] R. E. Quandt, "A New Approach to Estimating Switching Regressions," *J. Am. Stat. Assoc.* **67**, 306–310 (1972).
[36] S. B. Guthery, "Partition Regression," *J. Am. Stat. Assoc.* **69**, 945–947 (1974).
[37] A. R. Gallant and W. A. Fuller, "Fitting Segmented Polynomial Regression Models Whose Join Points Have to Be Estimated," *J. Am. Stat. Assoc.* **68**, 144–147 (1973).
[38] M. A. Fischler and R. C. Bolles, "Perceptual Organization and Curve Partitioning," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 100–105 (1986).
[39] Y. Kurozumi and W. A. Davis, "Polygonal Approximation by the Minimax Method," *Comput. Graph. Image Process.* **19**, 248–264 (1982).
[40] K. Ichida and T. Kiyono, "Segmentation of Plane Curves," *Syst. Comput. Controls* **6**, 27–36 (1975).
[41] K. Reumann and A. P. M. Witkam, "Optimizing Curve Segmentation in Computer Graphics," in *Proc. Int. Computer Symposium 1973*, A. Gunther et al., eds., North-Holland Pub. Co., Amsterdam, pp. 467–472 (1974).
[42] J. R. Rice, "Numerical Computation with General Two-Dimensional Domains," *ACM Trans. Math. Software* **10**, 443–452 (1984).
[43] T. Pavlidis and D. J. Sakrison, "Applications of a Simple Statistical Model for Curves," *IEEE Proc.* **IP 79**, 599–603 (1979).
[44] C. M. Williams, "Bounded Straight-Line Approximation of Digitized Planar Curves and Lines," *Comput. Graph. Image Process.* **16**, 370–381 (1981).
[45] R. L. Kashyap and B. J. Oomen, "Scale Preserving Smoothing of Polygons," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, 667–671 (1983).
[46] Y. Liao, "A Two-Stage Method of Fitting Conic Arcs and Straight-Line Segments to Digitized Contours," *IEEE Proc.* **IP 81**, 224–229 (1981).
[47] D. W. Bacon and D. G. Watts, "Estimating the Transition Between Two Intersecting Straight Lines," *Biometrika* **58**, 525–534 (1971).

Appendix A
WILLSKY AND JONES GLR FORMALISM

A.1 EXPLOITING LINEARITY

Willsky and Jones[17] provide without derivation a formalism they call the Generalized Likelihood Ratio (GLR) method for detecting and characterizing discontinuous (jump) inputs to linear systems. The elements of the GLR method are outlined in Section 2.3 and, for completeness, are derived here.

We assume that the jump detection problem is formulated as per Section 2.1. From Eq. 1,

$$x(k) = \Phi(k,k-1) \cdot x(k-1) + \Gamma(k-1) \cdot w(k-1) + \delta_{\theta,k} \cdot \nu \ , \tag{A-1}$$

from which it is obvious that $x(k)$ undergoes a jump when $k = \theta$.

As discussed in connection with Eqs. 15, and analogous to Eq. 98, we can write

$$x(k) = x_1(k) + x_2(k) \ , \tag{A-2}$$

where $x_1(k)$ is the value that $x(k)$ would take in the absence of a jump ($\nu = 0$ or $k < \theta$), and $x_2(k)$ is the state perturbation in response to a jump.

As first noted by McAulay and Denlinger,[1] the linearity of the system equations allows their solution in response to a jump excitation to be written as the sum of a jump-independent stochastic component (x_1) and a second component (x_2) linearly proportional to the jump amplitude. Moreover, with the jump amplitude and time taken as deterministic (though unknown) quantities, the jump-dependent component x_2 is seen to be deterministic.

From Eqs. 1, 2, and A-2,

$$x_1(k+1) = \Phi(k+1,k)x_1(k) + \Gamma(k)w(k) \tag{A-3}$$

$$z_1(k+1) = H(k+1)x_1(k+1) + \nu(k+1) \tag{A-4}$$

and

$$x_2(k+1) = \Phi(k+1,k)x_2(k) + \delta_{\theta,k+1} \cdot \nu \tag{A-5}$$

$$z_2(k+1) = H(k+1)x_2(k+1) \ . \tag{A-6}$$

Equation A-5 is subject to the initial conditions

$$x_2(\theta-1) = 0 \ . \tag{A-7}$$

A.2 SOLUTION FOR $x_2(k)$

Setting first $k = \theta - 1$ and then $k > \theta$ in Eq. A-5, and using Eq. A-7, we obtain

$$x_2(\theta) = \nu \tag{A-8}$$

and

$$x_2(k+1) = \Phi(k+1,k)x_2(k) \ , \quad k \geq \theta \ . \tag{A-9}$$

With the definition

$$x_2(k) \equiv \Phi(k,\theta) \cdot \nu \ , \tag{A-10}$$

it follows from Eqs. A-7 to A-10 that

$$\Phi(k,\theta) = 0 \ , \quad k < 0 \tag{A-11}$$

$$\Phi(\theta,\theta) = I \tag{A-12}$$

$$\Phi(k+1,\theta) = \Phi(k+1,k) \cdot \Phi(k,\theta) \ , \tag{A-13}$$

which appeared in the text previously as Eqs. 16 to 18 (Section 2.3).

Equations A-12 and A-13 can be solved by induction to obtain

$$\Phi(k+1,\theta) = \prod_{n=\theta}^{k} \Phi(n+1,n) \ . \tag{A-14}$$

From Eqs. A-6 and A-10,

$$z_2(k) = H(k) \cdot \Phi(k,\theta) \cdot \nu \ . \tag{A-15}$$

A.3 JUMP SIGNATURES

The linearity of the Kalman filter, Eqs. 5–14, allows us to decompose the state estimate, analogous to Eq. A-2, as

$$\hat{x}(k|n) = \hat{x}_1(k|n) + \hat{x}_2(k|n) \ . \tag{A-16}$$

From Eqs. 7 to 9 and 13, we have

$$\hat{x}_2(k+1|k) = \Phi(k+1,k)\hat{x}_2(k|k) \tag{A-17}$$

$$\hat{x}_2(k|k) = \hat{x}_2(k|k-1) + K(k)\gamma_2(k) \tag{A-18}$$

$$\gamma_2(k) = z_2(k) - H(k)\hat{x}_2(k|k-1) \ . \tag{A-19}$$

From Eqs. A-15 and A-19,

$$\gamma_2(k) = H(k) \cdot [\Phi(k,\theta)\nu - \hat{x}_2(k|k-1)] \ ,$$

which, with Eq. A-17, becomes

$$\gamma_2(k) = H(k) \cdot [\Phi(k,\theta)\nu - \Phi(k,k-1)\hat{x}_2(k-1|k-1)] \ . \tag{A-20}$$

With the definitions

$$\hat{x}_2(k|k) \equiv F(k;\theta) \cdot \nu \tag{A-21}$$

and

$$\gamma_2(k) \equiv G(k;\theta) \cdot \nu \ , \tag{A-22}$$

Eq. A-20 becomes

$$G(k;\theta) = H(k) \cdot [\Phi(k,\theta) - \Phi(k,k-1)F(k-1;\theta)] \ . \qquad \text{(A-23)}$$

Equation A-23 appeared previously as Eq. 23 in the text.

It can be shown from Eqs. A-17, A-18, and A-20 that

$$\hat{x}_2(k|k) = [I - K(k)H(k)]\Phi(k,k-1)\hat{x}_2(k-1|k-1) + K(k)H(k)\Phi(k,\theta)\nu \ . \qquad \text{(A-24)}$$

This can be written as

$$F(k;\theta) = A(k)F(k-1;\theta) + B(k) \ , \qquad \text{(A-25)}$$

where we define

$$F(k;\theta) \cdot \nu \equiv \hat{x}_2(k|k) \ , \qquad \text{(A-26)}$$

$$A(k) \equiv [I - K(k)H(k)]\Phi(k,k-1) \ , \qquad \text{(A-27)}$$

$$B(k) \equiv K(k)H(k)\Phi(k,\theta) \ . \qquad \text{(A-28)}$$

Defining the auxiliary variable $\Theta(k;\theta)$ as the homogeneous solution of Eq. A-25, i.e.,

$$\Theta(k;\theta) = A(k)\Theta(k-1;\theta) \ , \qquad \text{(A-29)}$$

with

$$\Theta(\theta;\theta) = I \ , \qquad \text{(A-30)}$$

it can be shown by back substitution into Eq. A-24 that the solution of Eq. A-25 is given by

$$F(k;\theta) = \sum_{n=\theta}^{k} \Theta(k;n)B(n) \ . \qquad \text{(A-31)}$$

Equations A-27 and A-29 lead directly to Eq. 25 in the text, and Eqs. A-28 and A-31 lead to Eq. 24.

From Eqs. A-24 and A-26,

$$F(k;\theta) = K(k)\{H(k)[\Phi(k,\theta) - \Phi(k,k-1)F(k-1;\theta)]\} + \Phi(k,k-1)F(k-1;\theta) \ ,$$

which, with Eq. A-23, may be written as

$$F(k;\theta) = K(k)G(k;\theta) + \Phi(k,k-1)F(k-1;\theta) \ . \qquad \text{(A-32)}$$

Equation A-32 appears as Eq. 49 in Ref. 17.

A.4 THE LIKELIHOOD RATIO

Jump detection is formulated as a hypothesis testing problem as follows:[17]

$$\begin{aligned} H_0&: \text{ Hypothesis that no jump has occurred, i.e., that } k < \theta \\ H_1&: \text{ Hypothesis that a jump has occurred, i.e., that } k \geq \theta \ , \end{aligned} \tag{A-33}$$

and, equivalently,

$$\begin{aligned} H_0&: \quad \gamma(k) = \gamma_1(k) \\ H_1&: \quad \gamma(k) = \gamma_1(k) + G(k;\theta) \cdot \nu \ , \end{aligned} \tag{A-34}$$

where $\gamma_1(k)$ is zero mean and spectrally white.

Maximum likelihood estimates for θ and ν, denoted $\hat{\theta}$ and $\hat{\nu}$, are obtained implicitly as the values of θ and ν that maximize the joint conditional density:

$$p[\gamma(1), \ldots, \gamma(k)|H_1,\theta,\nu] \ .$$

The choice between H_0 and H_1 is based on the criterion

$$\Lambda(k) \begin{cases} > \eta \Rightarrow H_1 \\ < \eta \Rightarrow H_0 \end{cases} \ , \tag{A-35}$$

where η is a threshold ultimately relatable to probabilities of false detection and missed detection, and the generalized likelihood ratio $\Lambda(k,\theta)$ is defined as

$$\Lambda(k,\theta) = \frac{p[\gamma(1), \ldots, \gamma(k)|H_1,\hat{\theta},\hat{\nu}]}{p[\gamma(1), \ldots, \gamma(k)|H_0]} \ . \tag{A-36}$$

Since the conditional densities in Eq. A-36 are both Gaussian, we can simplify the implementation of Eq. A-35 by taking the logarithm of both sides, to obtain

$$\ell_{k,\theta} = 2 \ln[\Lambda(k,\theta)] = \ell_1 - \ell_2 \ , \tag{A-37}$$

where

$$\ell_1 \equiv \sum_{n=\theta}^{k} \gamma_n^T \cdot V_n^{-1} \cdot \gamma_n \tag{A-38}$$

and

$$\ell_2 \equiv \sum_{n=\theta}^{k} (\gamma_n - G_{n,\theta} \cdot \nu)^T \cdot V_n^{-1} \cdot (\gamma_n - G_{n,\theta} \cdot \nu) \ . \tag{A-39}$$

For conciseness of notation, subscripts and arguments are used interchangeably in Eqs. A-38, A-39, and the rest of this section. For example, $G_{j,\theta} = G(j;\theta)$. Also, we recall that V_n in Eqs. A-38 and A-39 is the covariance of γ (cf. Eq. 11 in the text), while $G_{n,\theta} \cdot \nu$ in Eq. A-39 is the mean value of γ_n subject to hypothesis H_1.

Equation A-39 can be expanded to obtain

$$\ell_1 \equiv \sum_{n=\theta}^{k} \gamma_n^T \cdot V_n^{-1} \gamma_n + \sum_n \nu^T \cdot (G_{n,\theta}^T \cdot V_n^{-1} \cdot G_{n,\theta}) \cdot \nu - \sum_n (\gamma_n^T \cdot V_n^{-1} \cdot G_{n,\theta} \cdot \nu + \nu^T \cdot G_{n,\theta}^T \cdot V_n^{-1} \cdot \gamma_n) . \tag{A-40}$$

From Eqs. A-37, A-38, and A-40,

$$\ell_{k,\theta} = -\nu^T \cdot \left[\sum_n G_{n,\theta}^T \cdot V_n^{-1} \cdot G_{n,\theta} \right] \cdot \nu + \left[\sum_n G_{n,\theta}^T \cdot V_n^{-1} \cdot \gamma_n \right]^T \cdot \nu + \nu^T \cdot \left[\sum_n G_{n,\theta}^T \cdot V_n^{-1} \cdot \gamma_n \right] . \tag{A-41}$$

With the definitions

$$d_{k,\theta} \equiv \sum_n G_{n,\theta}^T \cdot V_n^{-1} \cdot \gamma_n \tag{A-42}$$

and

$$C_{k,\theta} \equiv \sum_n G_{n,\theta}^T \cdot V_n^{-1} \cdot G_{n,\theta} \tag{A-43}$$

and noting that

$$d_{k,\theta}^T \cdot \nu = \nu^T \cdot d_{k,\theta} , \tag{A-44}$$

we write Eq. A-41 as

$$\ell_{k,\theta} = -\nu^T \cdot C_{k,\theta} \cdot \nu + 2 d_\theta^T \cdot \nu . \tag{A-45}$$

Equations A-42 and A-43 were given previously in the text as Eqs. 29 and 30, respectively.

Maximum likelihood estimates for ν and θ, denoted $\hat{\nu}$ and $\hat{\theta}$, are obtained by setting equal to zero the appropriate partial derivative of Eq. A-45, as follows. We note that

$$\nabla_\nu (\nu^T \cdot C_{k,\theta} \cdot \nu) = 2 C_{k,\theta} \cdot \nu \tag{A-46}$$

and

$$\nabla_\nu (d_{k,\theta}^T \cdot \nu) = d_{k,\theta} . \tag{A-47}$$

From Eqs. A-45 to A-47,

$$\nabla_\nu (\ell_{k,\theta}) = -2 C_{k,\theta} \cdot \nu + 2 d_{k,\theta} . \tag{A-48}$$

However,

$$\nabla_{\nu}(\ell_{k,\theta})\Big|_{\substack{\nu=\hat{\nu}\\ \theta=\hat{\theta}}} = 0 \ . \tag{A-49}$$

From Eqs. A-48 and A-49,

$$\hat{\nu}_k = C_{k,\hat{\theta}}^{-1} \cdot d_{k,\hat{\theta}} \ , \tag{A-50}$$

which appeared in the text as Eq. 28.

From Eqs. A-45 and A-50,

$$\ell_{k,\theta} = -d^T (C^{-1})^T C\, C^{-1}\, d + 2d^T\, C^{-1}\, d \ .$$

However, since

$$C = C^T$$

and

$$C^{-1} = (C^{-1})^T \ ,$$

we obtain

$$\ell(k;\theta) = d^T(k;\theta) \cdot C^{-1}(k;\theta) \cdot d(k;\theta) \ , \tag{A-51}$$

which appeared in the text as Eq. 32.

Appendix B
ONE-PARAMETER RECURSIVE REGRESSION

In this Appendix we derive our one-parameter recursive regression algorithm, Eqs. 8′-14′.

From Eqs. 12′ and 14′,

$$P_s(j) = [1 - j^2 \cdot P_s(j-1)/V(k)] \cdot P_s(j-1) \ . \tag{B-1}$$

Substituting Eq. 11′ into Eq. B-1, where

$$V(k) = j^2 \cdot P_s(j-1) + \sigma^2 \ , \tag{11′}$$

we obtain

$$P_s(j) = \left[\frac{\sigma^2}{j^2 P_s(j-1) + \sigma^2}\right] \cdot P_s(j-1) \ . \tag{B-2}$$

Defining

$$p(j) \equiv P_s(j)/\sigma^2 \ , \tag{B-3}$$

it follows from Eq. B-2 that

$$p(j) = p(j-1)/[1 + j^2 \cdot p(j-1)] \ ,$$

which, upon inversion, becomes

$$[1/p(j)] = j^2 + [1/p(j-1)] \ . \tag{B-4}$$

With the definition

$$q(j) \equiv 1/p(j) \ , \tag{B-5}$$

Eq. B-4 becomes

$$q(j) = q(j-1) + j^2 \ . \tag{B-6}$$

Substituting the following trial solution into Eq. B-6,

$$q(j) = a \cdot j^3 + b \cdot j^2 + c \cdot j \ , \tag{B-7}$$

we can solve for the unknown coefficients (a,b,c), to obtain

$$(a,b,c) = \frac{1}{6} \cdot (2,3,1) \ . \tag{B-8}$$

From Eqs. B-5, B-7, and B-8,

$$p(j) = 6/[j(j+1)(2j+1)] \ . \tag{B-9}$$

From Eqs. B-9 and B-3,

$$P_s(j) = 6 \cdot \sigma^2/[j(j+1)(2j+1)] \ , \tag{B-10}$$

which appeared in the text as Eq. 64. From Eqs. B-3 and 11′,

$$V(k) = \sigma^2 \cdot [1 + j^2 \cdot p(j-1)] \ . \tag{B-11}$$

Substituting Eq. B-9 into Eq. B-11, we obtain

$$V(k) = \sigma^2 \cdot \left[\frac{j+1}{j-1} \cdot \frac{2j+1}{2j-1}\right] \ , \tag{B-12}$$

which appeared in the text as Eq. 63. From Eqs. B-10, B-12, and 12′,

$$K_s(j) = 6/[(j+1)(2j+1)] \ , \tag{B-13}$$

which appeared in the text as Eq. 62. From Eqs. 8′ and 9′,

$$\gamma(k) = z(k) - j \cdot \hat{s}(j-1) \ , \tag{B-14}$$

which appeared in the text as Eq. 61.

Equation 60 requires no derivation, having appeared previously as Eq. 13′:

$$\hat{s}(j) = \hat{s}(j-1) + K_s(j) \cdot \gamma(k) \ . \tag{B-15}$$

Equations B-15, B-14, B-13, B-12, and B-10, appearing in the text as Eqs. 60–64, provide the desired one-parameter recursive regression procedure.

Appendix C
SOLUTION FOR THE AUXILIARY VARIABLE

In this section we derive Eqs. 89 as the solutions of the difference Eqs. 88 subject to initial conditions given by Eq. 27.

With the definitions

$$A_j \equiv \Theta_{21}(j;r) \tag{C-1}$$

$$B_j \equiv \Theta_{22}(j;r) \tag{C-2}$$

$$C_j \equiv K_s(j) = 6/[(j+1)(2j+1)] \ , \tag{C-3}$$

Eqs. 88 are written as

$$A_j = (1 - j \cdot C_j) \cdot A_{j-1} - C_j \tag{C-4}$$

$$B_j = (1 - j \cdot C_j) \cdot B_{j-1} \tag{C-5}$$

From Eq. C-3,

$$(1 - j \cdot C_j) = \frac{(j-1)(2j-1)}{(j+1)(2j+1)} \ . \tag{C-6}$$

From Eqs. C-5 and C-6,

$$(j+1)(2j+1) \cdot B_j = j(2j-1) \cdot B_{j-1} - (2j-1) \cdot B_{j-1} \ . \tag{C-7}$$

With the definition

$$D_j \equiv j(2j-1) \cdot B_{j-1} \ , \tag{C-8}$$

Eq. C-7 is written as

$$j \cdot D_{j+1} = (j-1) \cdot D_j \ ,$$

$$\Theta_{21}(r;r) = 0 \ , \tag{C-20}$$

Eq. C-19 becomes

$$\Theta_{21}(j;r) = \frac{3[r(r+1) - j(j+1)]}{j(j+1)(2j+1)} \ , \tag{C-21}$$

which appeared in the text as Eq. 89a.

which has the solution

$$(j-1) \cdot D_j = \text{constant} \ . \tag{C-9}$$

From Eqs. C-2, C-8, and C-9,

$$\Theta_{22}(j;r) = \frac{\text{constant}}{j(j+1)(2j+1)} \ . \tag{C-10}$$

Obtaining the appropriate initial condition from Eq. 27,

$$\Theta_{22}(r;r) = 1 \ , \tag{C-11}$$

Eq. C-10 becomes

$$\Theta_{22}(j;r) = \frac{r(r+1)(2r+1)}{j(j+1)(2j+1)} \ , \tag{C-12}$$

which appeared in the text as Eq. 89b.

From Eqs. C-3, C-4, and C-6,

$$(j+1)(2j+1) \cdot A_j = (j-1)(2j-1) \cdot A_{j-1} - 6 \ . \tag{C-13}$$

Defining

$$E_j \equiv (j-1)(2j-1) \cdot A_{j-1} \ , \tag{C-14}$$

Eq. C-13 becomes

$$(j+1) \cdot E_{j+1} = j \cdot E_j - 6j \ . \tag{C-15}$$

Defining

$$F_j \equiv j \cdot E_j \ , \tag{C-16}$$

Eq. C-15 becomes

$$F_{j+1} = F_j - 6j \ . \tag{C-17}$$

We can show by back substitution that the solution of Eq. C-17 is given by

$$F_j = a - 3j(j - 1) \ , \tag{C-18}$$

where a is a constant.

From Eqs. C-1, C-14, C-16, and C-18,

$$\Theta_{21}(j;r) = \frac{a - 3j(j + 1)}{j(j + 1)(2j + 1)} \ . \tag{C-19}$$

Obtaining the initial condition from Eq. 27,

Appendix D
NUMERICAL EXPERIMENTS AT 7:1 DATA COMPACTION

In Section 6.4 we presented graphs depicting polyline fit, fitting error, and histogrammed fitting error, for the GLR polyline algorithm. Here we provide results in an identical format for the following additional algorithms:

EMK	50-knot approximation (Fig. D-1)
CIM	50-knot approximation (Fig. D-1)
DP	49-knot approximation (Fig. D-2)
LSBPT	50-knot approximation (Fig. D-3)
HOP-S	51-knot approximation (Fig. D-4)
BAD	50-knot approximation (Fig. D-5)
RLS1	50-knot approximation (Fig. D-6)
RLS2	50-knot approximation (Fig. D-6)

Not all algorithms were able to provide exactly a 50-knot approximation; consequently, the actual numbers of knots are given above for the fit shown in the figure. This also explains the discrepancy between some of the performance statistics appearing below the histograms in the figures and the corresponding results in Table 3 (in the main text). If a particular algorithm could not obtain the desired 50-knot approximation, the statistics in Table 3 are an average for fits of 49 and 51 knots.

Fits provided by algorithms EMK and CIM were identical in this instance and so are provided as a single figure. Similarly, results for RLS1 and RLS2 were identical for the 50-knot case and are therefore represented by a single figure.

References to the literature and definitions of the algorithm acronyms are provided in Table 2 (in the main text).

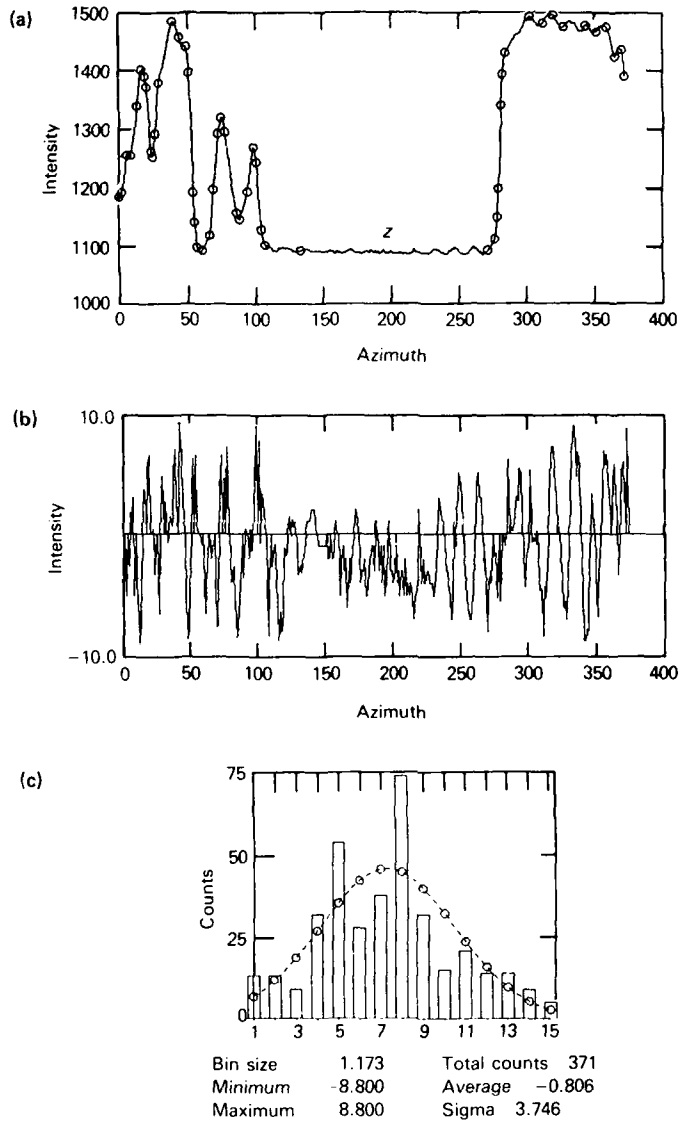


Figure D-1 EMK and CIM polyline algorithm evaluations. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

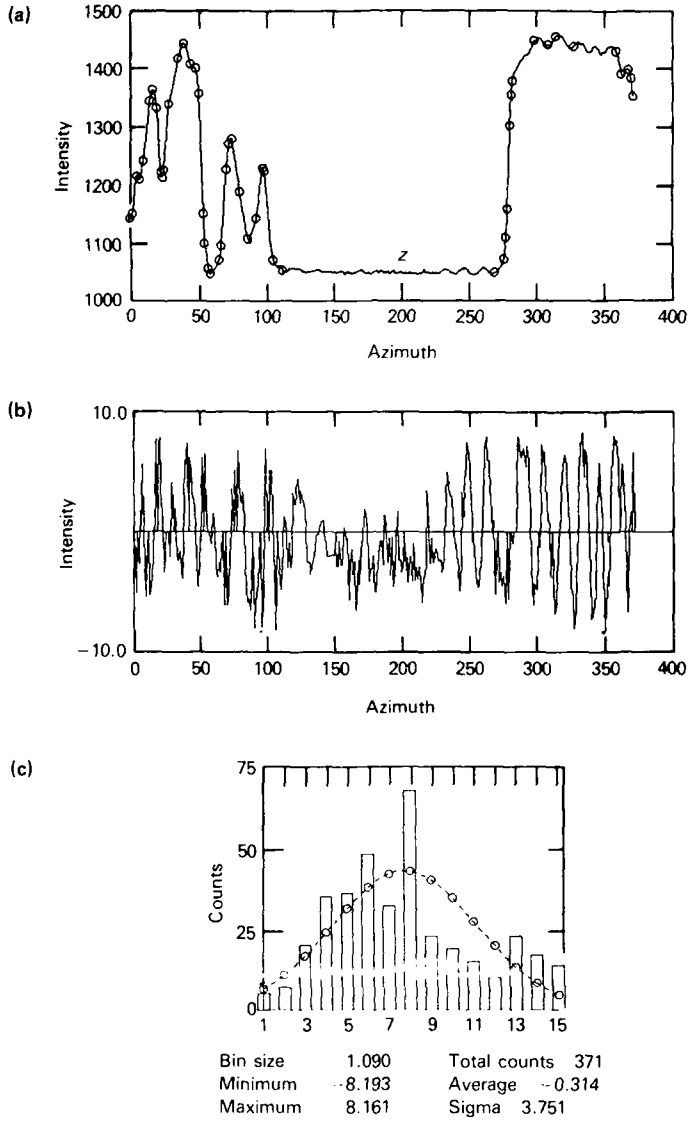


Figure D-2 DP polyline algorithm evaluation. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

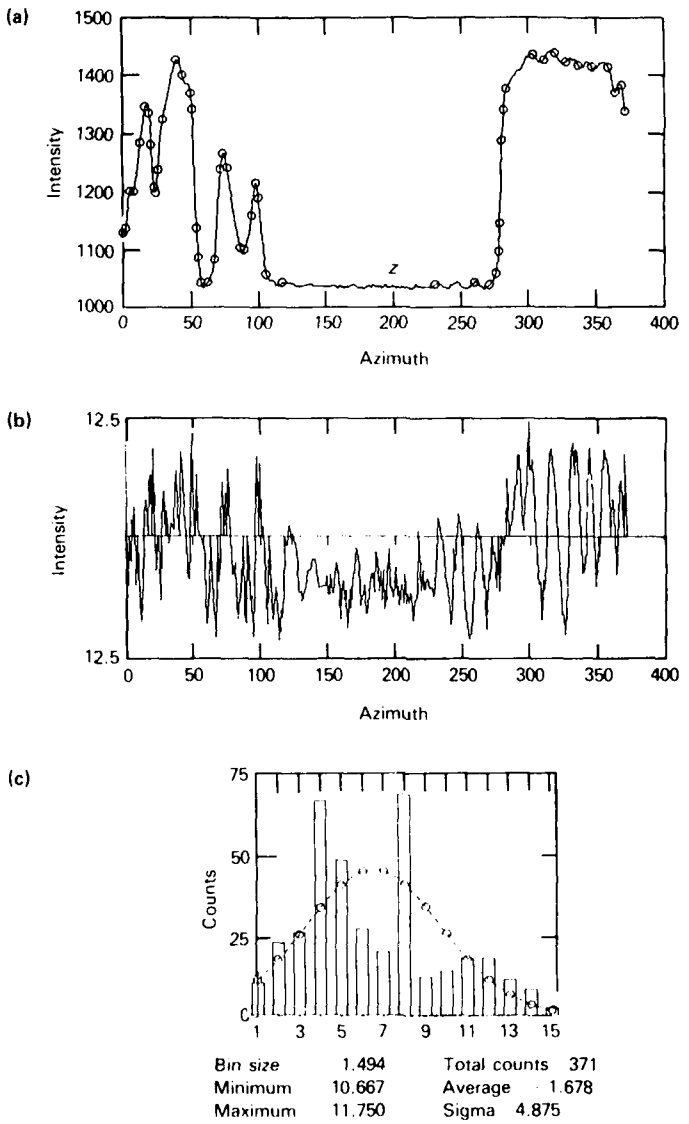


Figure D-3 LSBPT polyline algorithm evaluation. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

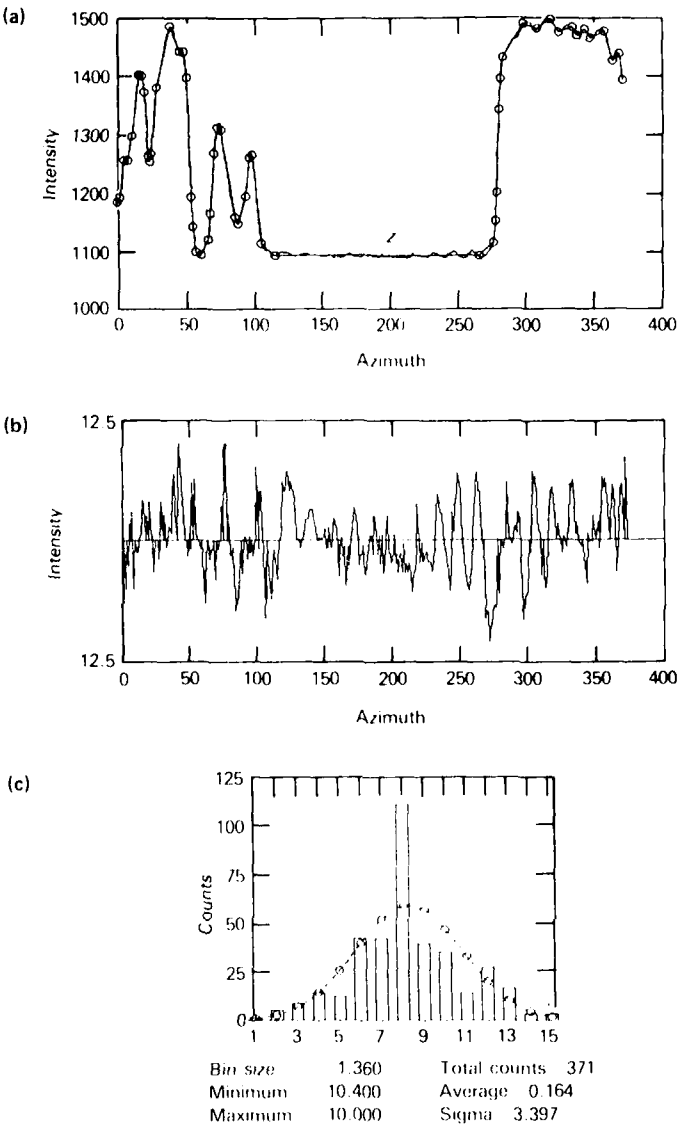


Figure D-4 HOP-S polyline algorithm evaluation. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

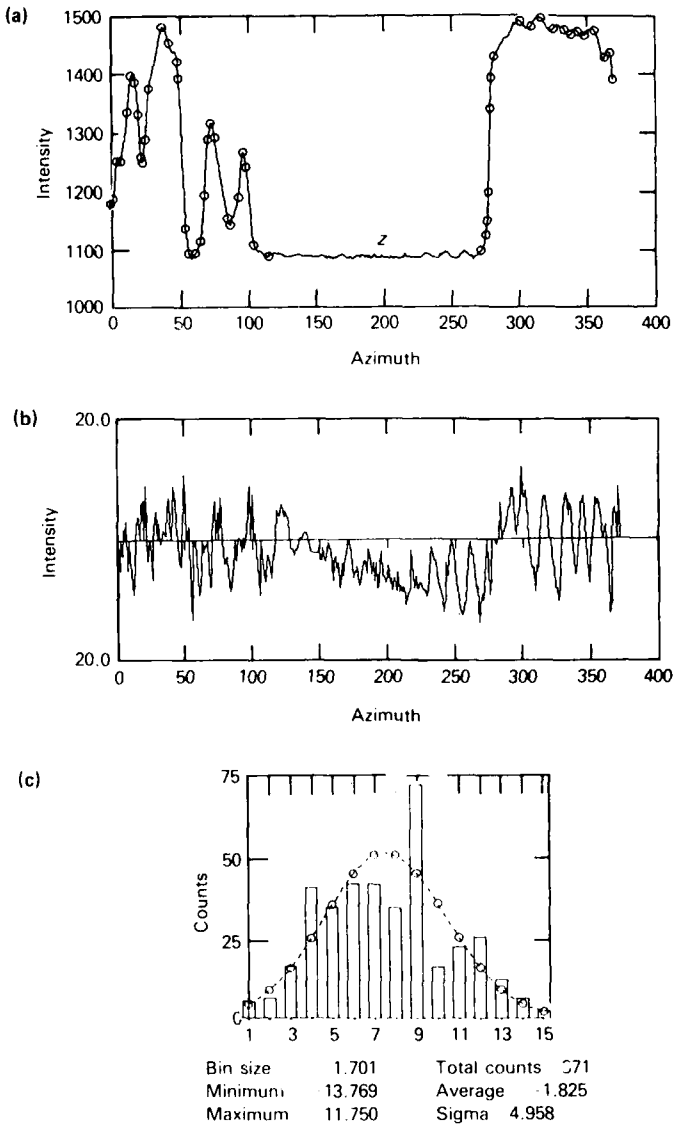


Figure D-5 BAD polyline algorithm evaluation. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

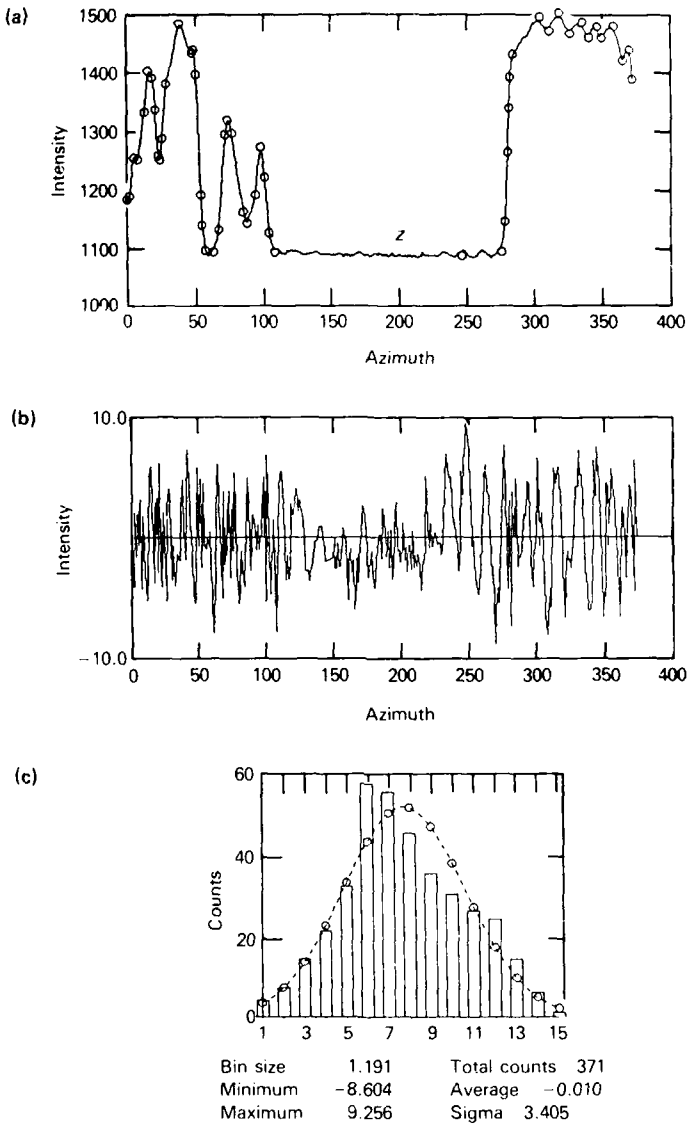


Figure D-6 RLS1 and RLS2 polyline algorithm evaluation. (a) IR cloud waveform (solid curve, *z*) and polyline knots (o). (b) Pointwise difference between *z* and polyline approximation. (c) Histogram of (b) with fitted Gaussian density (circles and dashed curve).

Appendix E
PERFORMANCE STATISTICS VERSUS DATA COMPACTION

In Section 6.6 we presented curves of performance statistics versus data compaction for the GLR, RLS1, and CIM polyline algorithms. Here we provide results in an identical format for an additional set of algorithms:

EMK	(solid curves) versus CIM (dashed curves) (Fig. E-1)
DP	(solid curves) versus CIM (dashed curves) (Fig. E-2)
LSBPT	(solid curves) versus CIM (dashed curves) (Fig. E-3)
HOP-S	(solid curves) versus CIM (dashed curves) (Fig. E-4)
BAD	(solid curves) versus CIM (dashed curves) (Fig. E-5)
RLS2	(solid curves) versus CIM (dashed curves) (Fig. E-6)
RLS1	(solid curves) versus RLS2 (dashed curves) (Fig. E-7)

References to the literature and definitions of the algorithm acronyms are provided in Table 2 (in the main text).

Of the various algorithms described in the literature, CIM provides perhaps the best combination of good fitting accuracy, fast execution speed, robustness, and ease of use. For this reason, the performance of the other algorithms is generally compared with that of the CIM baseline. The exception to this is Fig. E-7, where we compare the two recursive least squares (RLS) algorithms developed from analytical considerations in Section 4.

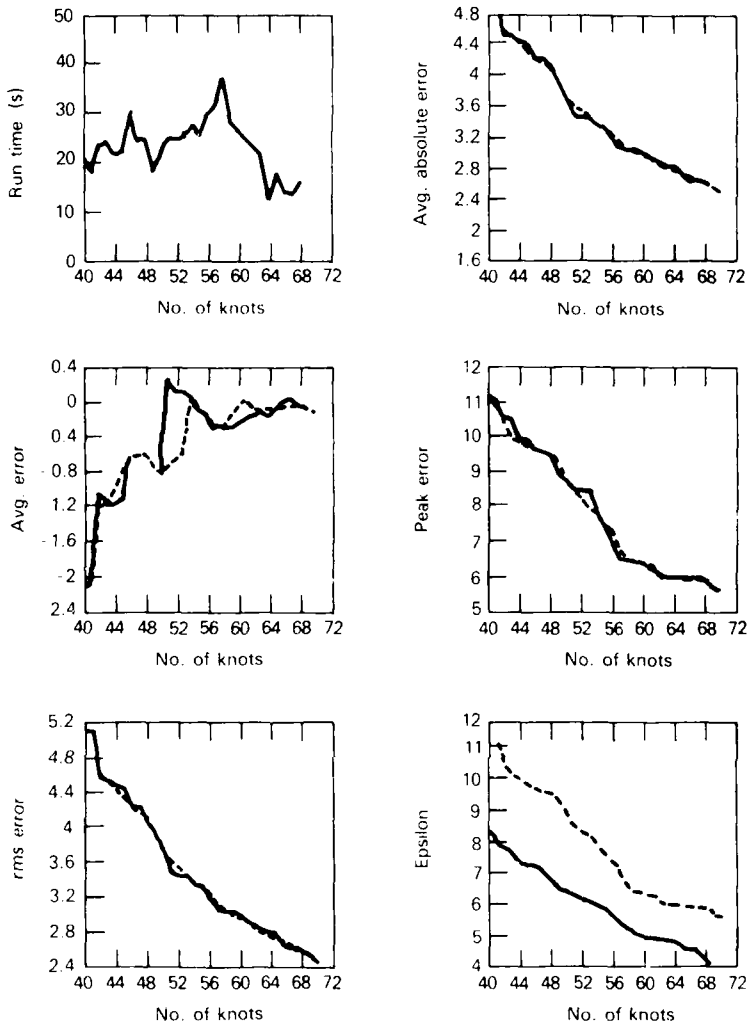


Figure E-1 Performance statistics as functions of data compaction for EMK (solid curves) and CIM (dashed curves).

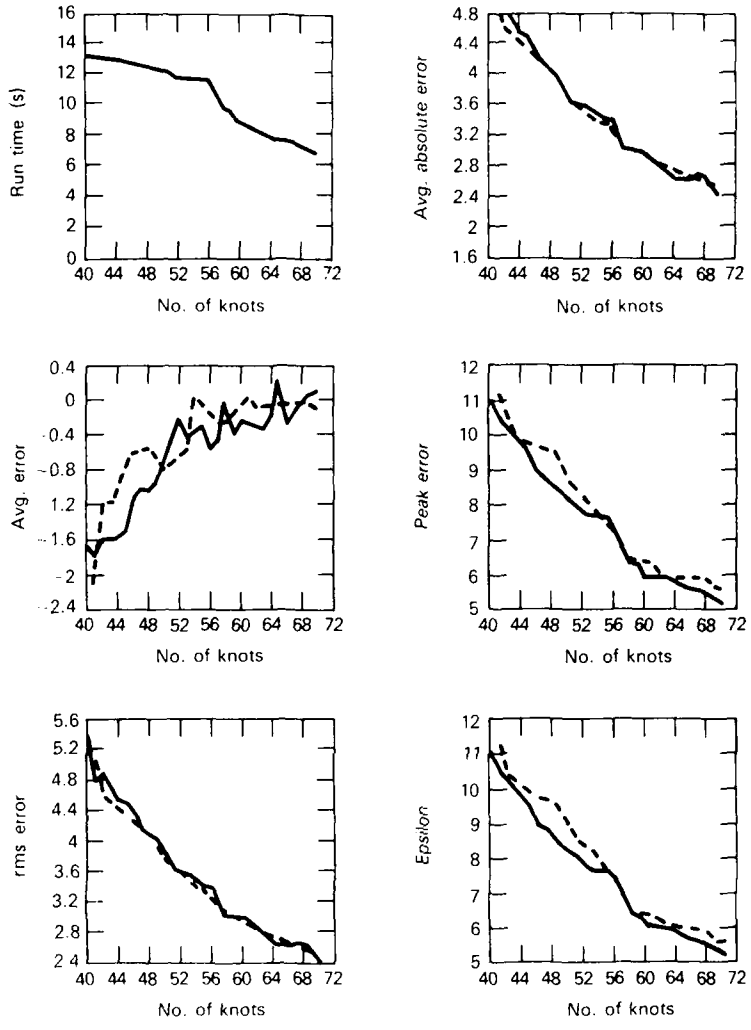


Figure E-2 Performance statistics as functions of data compaction for DP (solid curves) and CIM (dashed curves).

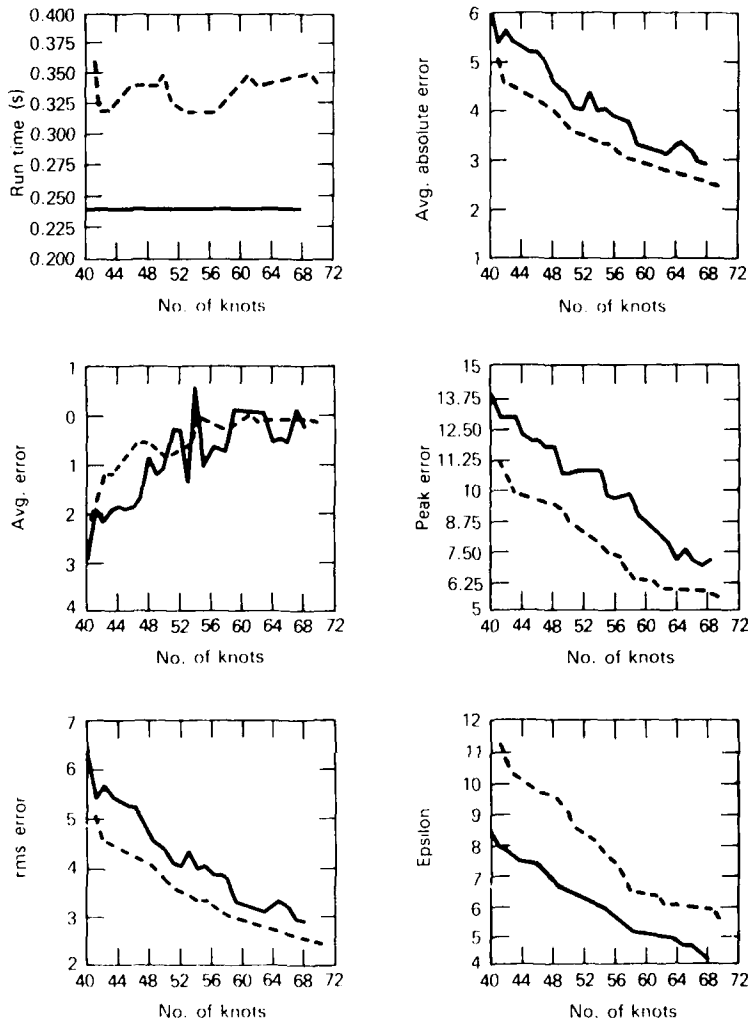


Figure E-3 Performance statistics as functions of data compaction for LSBPT (solid curves) and CIM (dashed curves).

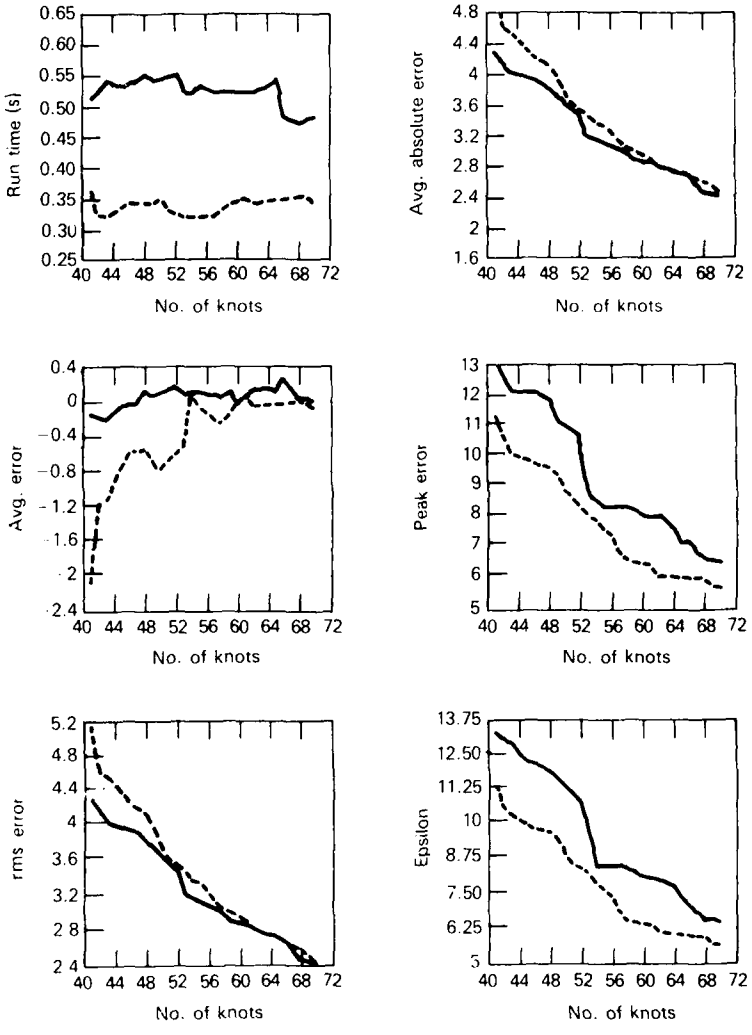


Figure E-4 Performance statistics as functions of data compaction for HOP-S (solid curves) and CIM (dashed curves).

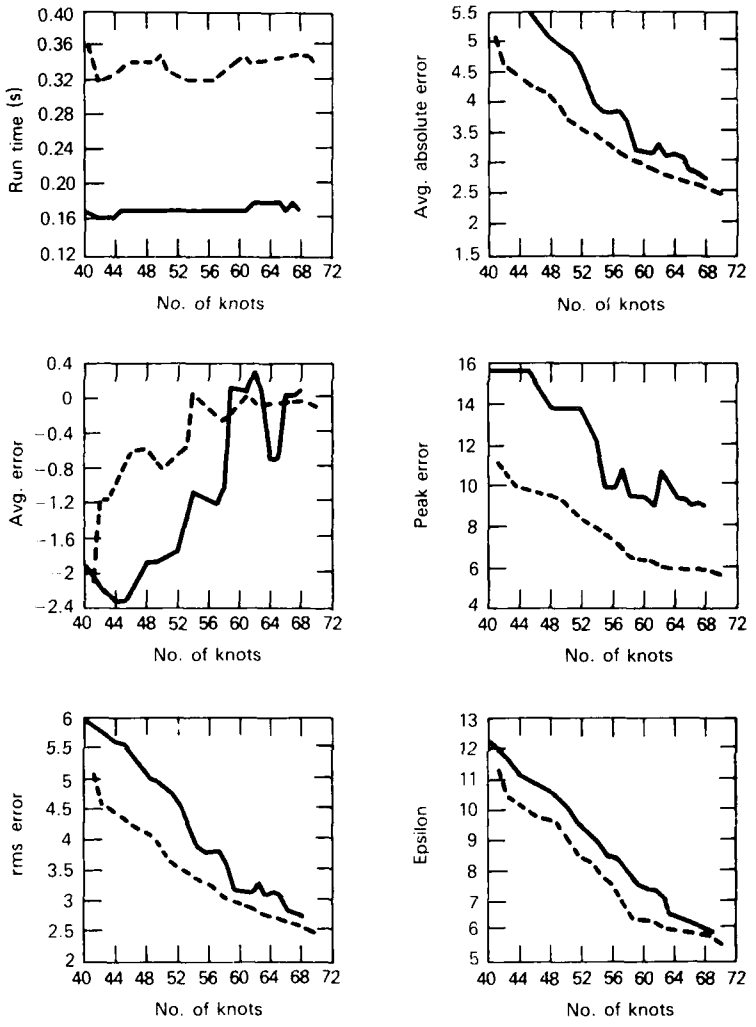


Figure E-5 Performance statistics as functions of data compaction for BAD (solid curves) and CIM (dashed curves).

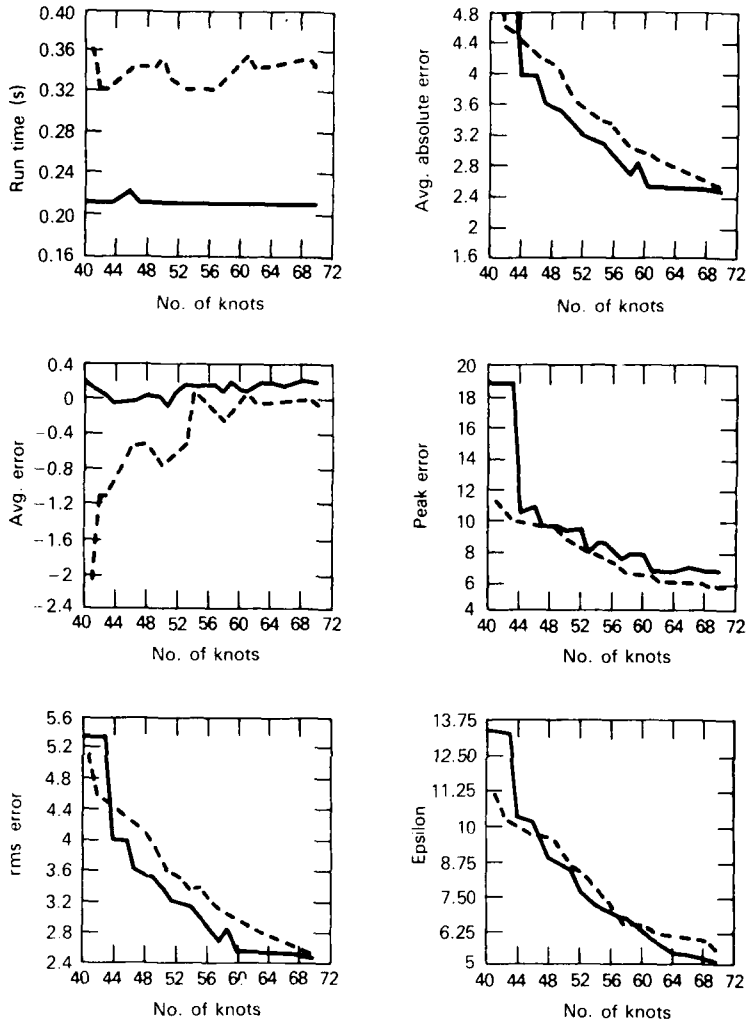


Figure E-6 Performance statistics as functions of data compaction for RLS2 (solid curves) and CIM (dashed curves).

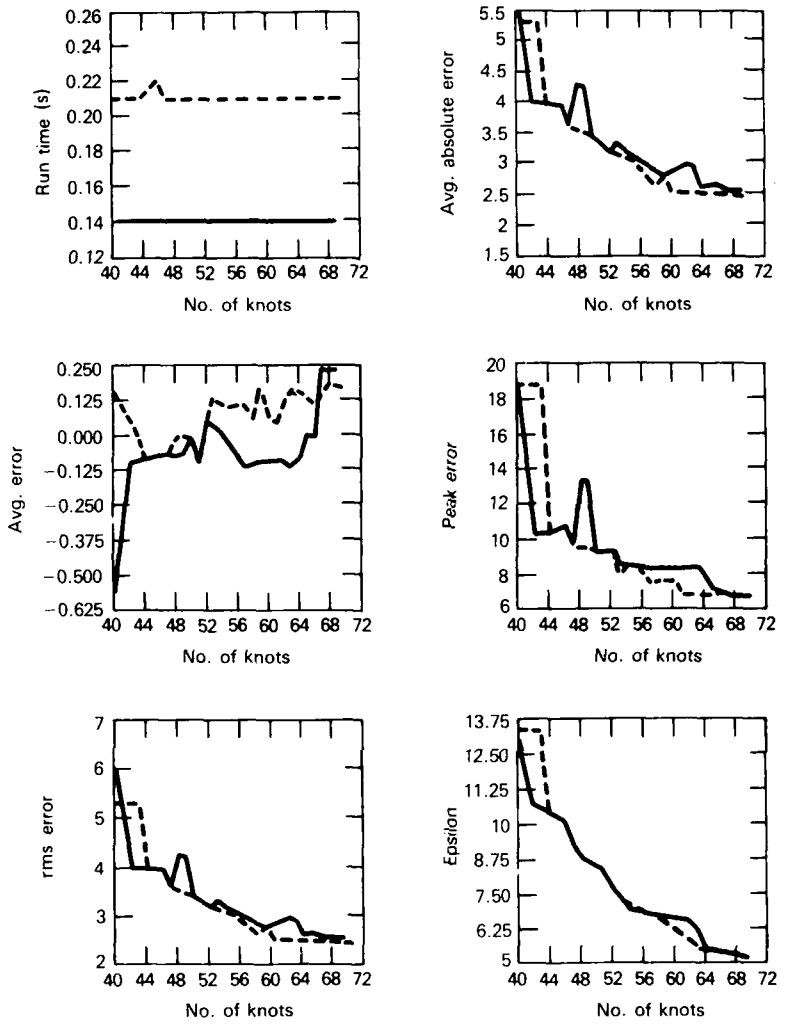


Figure E-7 Performance statistics as functions of data compaction for RLS1 (solid curves) and CIM (dashed curves).

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

The work in TG 1370 was done under Navy Contract N00039-87-C-5301 and is related to Task X8FA, supported by the JHU/APL Independent Research and Development Fund.

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF DEFENSE			
Secretary of Defense	Washington, DC 20301	G. C. Kopesak, OUSDRE (ET)	1
Office of the Under Secretary of Defense, Research and Engineering	Washington, DC 20301	J. MacCallum	1
Defense Technical Information Center	Alexandria, VA 22314	Accessions	12
Defense Advanced Research Projects Agency	Arlington, VA 20331	S. Karp	1
DEPARTMENT OF THE NAVY			
Chief of Naval Operations	Washington, DC 20350	OP 98	1
		OP 983	1
		OP 987	1
		OP 987B	2
		OP 35	1
		OP 35E	1
		OP 352L	1
		OP 355W	1
		OP 507D	1
Office of the Assistant Secretary of the Navy	Washington, DC 20350	E. Donaldson	1
		C. Kincaid	1
		R. E. Metry	1
		R. L. Rumpf	1
Office of Naval Research	Arlington, VA 22217	W. C. Condell, Jr., ONR-421	1
		E. Wegman, ONR-411SP	1
Office of Naval Technology	Arlington, VA 22209	ONT 07C	1
		ONT 0712	1
		ONT 0713	1
Naval Air Systems Command	Washington, DC 22202	AIR 320	1
		AIR 320B	1
		AIR 320D	1
		AIR 320R	1
		AIR 340J	1
		AIR 06	1
		AIR 0623	1
		Library, AIR 7226	2
		B. Dillon	1
		G. Savoca	1
Naval Air Development Center	Warminster, PA 18974	S. Campana, 3011	1
Naval Electronic Systems Command	Arlington, VA 20360	D. M. Ferreira	1
Naval Postgraduate School	Monterey, CA 92940	T. Tao	1
Naval Research Laboratory	Washington, DC 20375	T. G. Giallorenzi, 6500	1
		J. Kershenstein, 6520	1
		E. J. Stone, 6520	
Space & Naval Warfare Systems Command	Washington, DC 20360	Library	1
Naval Sea Systems Command	Washington, DC 22202	SEA 06	1
		SEA 06A	1
		SEA 06AT	1
		SEA 06AX	1
		SEA 06P	1
		SEA 06R	1
		SEA 62	1
		SEA 62B	1

Requests for copies of this report from DoD activities and contractors should be directed to DTIC. Cameron Station, Alexandria, Virginia 22314 using DTIC Form 1 and, if necessary, DTIC Form 55.

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF THE NAVY (continued)			
		SEA 62R	2
		SEA 62R1	2
		SEA 62R11	1
		SEA 62R2	1
		SEA 62R4	1
		SEA 62R5	2
		SEA 62R52	13
		SEA 62Z	1
		SEA 62Z2	1
		SEA 62Z3	1
		SEA 62Z3B	1
		SEA 62Z31	2
		SEA 62Z31F	1
		Library, SEA 9961	2
		PMS 400	1
		PMS 400B	1
Naval Surface Weapons Center	Dahlgren, VA 22448	DL D2W	1
		DL G21	1
		DL G22	1
		DL G24	1
		Library	2
Naval Surface Warfare Center	White Oak, MD 20903	WO CR42	1
		WO G40	1
		WO K22	1
		WO N12	1
		B. Kessler	1
		C. W. Larson	1
		Library	2
Naval Weapons Center	China Lake, CA 93555	3906	2
		3921	1
		3943	1
		L. Wilkins, 39	1
		Library	2
Pacific Missile Test Center	Pt. Mugu, CA 93042	4045	1
U.S. Naval Academy	Annapolis, MD 21402	Director of Research	2
NAVPRO	Laurel, MD 20707		1
DEPARTMENT OF THE ARMY			
Ballistic Missile Defense Advanced Technology Center	Huntsville, AL 35807	ATC-R	1
		P. Boyd, ATC-RN	1
		HNV	1
		LEH	1
Missile Command	Huntsville, AL 35898	DRSMI OD	1
		DRSMI REI	1
		DRSMI RR	1
Night Vision and Electro-Optics Lab.	Fort Belvoir, VA 22060	J. A. Ratches	1
		W. R. Lawson	1
DEPARTMENT OF THE AIR FORCE			
Air Force Armament Division	Eglin AFB, FL 32542	AD/XRG	1
Air Force Avionics Laboratory	Wright-Patterson AFB, OH 45433	R. Nieter	1
		R. Sanderson, WRP-1	1
		Library	1
Aeronautical Systems Division	Wright-Patterson AFB, OH 45433	YYM	1

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
CONTRACTORS			
Aerodyne Research	Billerica, MA 01821	J. Draper	1
Aerospace Corporation	Los Angeles, CA 90009	P. Mazaika	1
		I. Spiro	1
Charles Stark Draper Laboratory	Cambridge, MA 02139	T. H. Brooks	1
Environmental Research Institute of Michigan	Ann Arbor, MI 48107	J. R. Maxwell	1
General Electric	Syracuse, NY 13221	A. F. Milton	1
General Research Corp.	McLean, VA 22101	R. Zirkind	1
Grumman Aerospace	Bethpage, NY 11714	P. E. Barry, A08-35	1
		J. Krassner	1
Honeywell, Inc.	Minneapolis, MN 55440	P. M. Narendra	1
		A. Cox	1
Hughes Aircraft Co.	Canoga Park, CA 91340	J. T. Hall	1
		R. L. Sendall	1
IBM Federal Systems Division	Westlake Village, CA 91361	J. C. Day	1
Institute for Defense Analyses	Alexandria, VA 22311	I. W. Kaye	1
		R. R. Legault	1
Lockheed Missile and Space Co.	Palo Alto, CA 94304	L. H. Wald	1
		N. Kulgein	1
Lou Massa	New York, NY 10021		
MIT Lincoln Laboratory	Lexington, MA 02173	H. Kleiman	1
Nichols Research Corp.	Huntsville, AL 35802	R. J. Nichols	1
North Carolina State University	Raleigh, NC 27709	W. E. Snyder	1
Ontar Corp.	Brookline, MA 02146	J. Schroeder	1
Opti Metrics	Ann Arbor, MI 48104	R. Meredith	1
Optical Science Co.	Placentia, CA 92670	D. Fried	1
Polytechnic Institute of New York	Farmingdale, NY 11735	A. Papoulis	1
Rand Corporation	Santa Monica, CA 90406	L. G. Mundie	1
Raytheon Missile Systems Division	Bedford, MA 01730	I. Goldstein	1
Riverside Research Institute	New York, NY 10036	R. G. Cestaro	1
Rockwell International	Downey, CA 90241	M. F. Sentovich	1
Science Applications	Ann Arbor, MI 48104	R. Turner	1
Sensor Systems Group	Waltham, MA 02154	M. Anapol	1
Spectral Sciences, Inc.	Burlington, MA 01803	L. S. Bernstein	1
SRI International	Menlo Park, CA 94025	J. Malick	1
Syn Optics Associates	Long Beach, CA 90803	R. Donohue	1
Visidyne Co.	Burlington, MA 01803	T. F. Zehnpfennig	1
W. J. Schafer Associates	Arlington, VA 22209	D. H. Leslie	1
Whittier College	Whittier, CA 90608	R. Rubin	1